8

ANALYSIS OF LANGUAGES FOR MAN-MACHINE VOICE COMMUNICATION

Robert Gary Goodman
May, 1976

See 1473 in back

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is
approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

ANALYSIS OF LANGUAGES FOR MAN-MACHINE VOICE COMMUNICATION

A DISSERTATION

SUBMITTED TO THE COMPUTER SCIENCE DEPARTMENT

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

by

Robert Gary Goodman
Carnegie-Mellon University

~~May 1976~~
~~Reprinted~~ September 1976

This work was supported by the Defense Advanced Research Projects Agency under contrac[illegible]4620-73-C-0074 and monitored by the Air Force Office of Scientific Re[illegible]arch.

ANALYSIS OF LANGUAGES
FOR
MAN-MACHINE VOICE COMMUNICATION

Robert Gary Goodman, Ph.D.
Stanford University, 1976

Comparing the relative performances of speech understanding systems has always been difficult and subject to speculation. Different tasks naturally require different vocabularies with varying acoustic similarities. Moreover, constraints imposed by the syntax may make recognition easier, even for vocabularies with high ambiguity. This thesis presents an analysis of ambiguity, restriction and complexity in speech understanding system languages. The ambiguity considered involves the similarity of acoustic signals and the ambiguity it causes at other levels of recognition. Phonemes spoken in isolation are misrecognized by both man and machine. Words and phrases having similar phonetic structure are confused. This confusion increases the complexity with connected speech but syntactic and other higher levels of knowledge provide additional constraints to reduce the ambiguity. This thesis examines ambiguity and complexity at the phonetic, lexical and syntactic levels. Ambiguity may also occur at the semantic and user discourse levels. The concepts presented here can be extended to these levels.

Measures are developed which permit the relative comparison of the difficulties of a given set of recognition tasks. We present notions of equivalent vocabulary size, branching factor, effective branching factor, search space size and search space reduction. All of these are useful as relative comparison measures. Briefly, the plan of research is to investigate, in order: phonetic ambiguity, word ambiguity, lexical ambiguity, syntactic constraint and the combined effects of lexical ambiguity and syntactic constraint.

First, the major source of ambiguity, the acoustic speech signal itself, is considered. Several measures for quantifying phonetic ambiguity are investigated and compared. These measures provide a basis for the computation of lexical and phrasal ambiguity.

A model for lexical ambiguity is presented which utilizes the knowledge of phonetic ambiguity and a general representation of the vocabulary to estimate the probability that an acoustic realization of some sequence of idealized phonemes will result in incorrect recognition. The average expected number of words retrieved in an syntactically unconstrained lexical search is computed from these probabilities. This number is called the equivalent size of the vocabulary. The 10 digits, for instance, have an equivalent size of 1.19 words, while the equivalent size of the spoken alphabet ("a", "b", . . . "z") is 3.87.

The syntax of languages for speech understanding systems imposes restrictions on the number of word pairs, triples, etc. which can occur in the language. These limitations can dramatically reduce the total size of the search space. One of the languages investigated has a 250 word vocabulary and an average sentence length of

8 words. Syntactic restrictions reduce the branching factor to 7.3. That is, on the average, one must disambiguate among 7 words.

Equivalent vocabulary size may be viewed as a branching factor in the case where there are no syntactic constraints. Thus, lexical ambiguity and syntactic restriction are measured in the same terms. This unification allows combined effects of vocabulary ambiguity and syntactic complexity to also be viewed as a branching factor. Two models for complexity of connected speech are defined. A "best" behavior model which assumes that word boundaries are known and therefore the only confusions that may arise are when two (or more) phonetically similar words have the same contexts. The effective branching factor obtained can be viewed as an optimistic representation of the expected behavior of the system. A "worst" case model is also discussed.

The important contribution of this thesis is that it provides a way to characterize the relative difficulties and accomplishments of different speech understanding systems. Vocabulary size is not a good measure of lexical complexity; some other measure of vocabulary size, normalized for relative ambiguity would be better. The number of production rules is not a useful measure of grammatical complexity. In fact, quite the opposite may be true; more rules imply more constraint. Some other measure, such as the average number of alternatives at each choice point would be better.

ACKNOWLEDGEMENTS

It is a great pleasure to have this opportunity to express my appreciation to Raj Reddy for his constant technical guidance and moral support. He has been a true friend and an excellent adviser.

I am grateful to my thesis committee: John McCarthy, Arthur Samuel, and Cordell Green for their helpful comments concerning this thesis. In addition, the many interesting discussions with my colleagues Lee Erman, Fredrick Hayes-Roth, Ellis Cohen and John McDermott were of great value to the success of this research.

I am deeply indebted to Mike Kelly for his personal interest in the completion of this work.

This dissertation is dedicated to my wife, Ann, whose help and encouragement have been invaluable.

This work was begun at Stanford University and continued at Carnegie-Mellon University. I am grateful to the members of both Computer Science Departments for providing assistance, support and outstanding research facilities. This work was also supported by the Advanced Research Projects Agency of the Department of Defense.

Special thanks go to Brian Reid for his valuable and patient assistance during the preparation of this thesis and to Bruce Lowerre for his expert programming support.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Comparing the relative performances of speech understanding systems has always been difficult and subject to speculation. Different tasks naturally require different vocabularies with varying acoustic similarities. Moreover, constraints imposed by the syntax may make recognition easier, even for vocabularies witn high ambiguity. This thesis presents an analysis of ambiguity, restriction and complexity in speech understanding system languages. The ambiguity considered involves the similarity of acoustic signals and the ambiguity it causes at other levels of recognition. Phonemes spoken in isolation are misrecognized by both man and machine. Words and phrases having similar phonetic structure are confused. This confusion increases the complexity with connected speech but syntactic and other higher levels of knowledge provide additional constraints to reduce the ambiguity. In this thesis, we will examine ambiguity and complexity at the phonetic, lexical and syntactic levels. Ambiguity may also occur at the semantic and user discourse levels. We believe that the concepts presented here can be extended to these levels in an analogous manner.

1.1. Ambiguity in Speech Understanding Systems

To illustrate some of the issues relating to complexity, consider the first two vocabularies shown in figure 1-1. The first vocabulary is the spoken letters "B", "D" and "V", while the second is comprised of the three digits "ONE", "TWO" and "THREE". It would not be difficult to elicit opinions as to which of these two vocabularies would be easier to recognize; and, most likely, there would be a consensus. In this case, intuition has given the correct answer. Consider now, vocabulary 3 which contains the spoken letters "A", "B" and "C". Would vocabulary 2 be easier to recognize than

Vocabulary 1:

"B"	<B>	<IY>
"D"	<D>	<IY>
"V"	<V>	<IY>

Vocabulary 2:

"ONE"	<W>	<AN>	<N>				
"TWO"	<T>	<IH>	<UW>				
"THREE"	<TH>	<ER>	<IY>	or	<TH>	<R>	<IY>

Vocabulary 3:

"A"	<EH>	<IH>	or	<EH>	<IX>
"B"	<B>	<IY>			
"C"	<S>	<IY>			

Figure 1-1. Some simple vocabularies with intuitive complexities.

vocabulary 3? Again, opinions are easy to come by, but, in this case there may not be agreement. An example of the performance of an isolated word recognition system will serve to illustrate that the number of words in a vocabulary may not be indicative of its complexity. Itakura[1975], in his word recognition system, investigated two vocabularies. The first vocabulary, called the alpha-digit vocabulary, contains the 26 letters of the English alphabet and the ten digits. The other is a vocabulary of 250 Japanese geographical names. The results, shown in figure 1-2, were that Itakura achieved 88.6% recognition for the alpha-digit vocabulary and 97.3% with the geographical names. Why is it that the alphabet and digits are more difficult to recognize than the 250 names? One might guess that the names were multi-syllabic and phonetically dis-similar. While Itakura did not list the names, it was stated that there were 3.5 syllables per word, on the average. The questions raised here will be answered in chapter 3.

To illustrate the effects of syntactic constraint, consider the results of Baker and Bahl[1975], also shown in figure 1-2. The languages used were telephone numbers and the "New Raleigh" language with vocabulary sizes of 10 words and 250 words, respectively. The recognition rates for these two tasks are roughly the same even though one has 25 times as many words as the other. The reason could be because the 250 word vocabulary is unambiguous or it could be due to the constraint imposed by the syntax. A more precise answer will appear in later chapters.

In this thesis we want to develop some measures which will permit the relative comparison of the difficulties of a given set of recognition tasks. We will present notions of equivalent vocabulary size, branching factor, effective branching factor, search space size and search space reduction. All of these are useful as relative comparison measures.

ISOLATED WORDS (Itakura, 1975)

Vocabulary	Recognition Rate(%)	Rejection Rate(%)	Error Rate(%)
Alpha-Digit	88.6	0	11.4
Japanese Geographical Names	97.3	1.7	1.0

CONSTRAINED LANGUAGES (Baker and Bahl, 1975)

	% Correct Words	% Correct Sentences
Telephone Numbers (7 decimal digits)	97.4	89
"New Raleigh Language"	98.3	81

Figure 1-2. Illustrations of comparative recognition rates.

1.2. Previous Research

Virtually every speech understanding system faces the problem of phonetic ambiguity; thus, there are many metrics which attempt to measure the similarity/difference of acoustic events. We have chosen the minimum prediction residual metric[Itakura, 1975] for use in this thesis. This metric is a measure of the distance or dissimilarity between segments of discrete-time signals.

There is considerable phoneme confusion data for human perception. Vowels[Petersen and Barney, 1952; Ladeforged and Broadbent, 1957]. Consonants[Miller and Nicely, 1955]. The Miller and Nicely paper discusses some theoretic concepts of information content of various distinguishable characteristics of the perception.

There is no known previous work in lexical ambiguity, except that in the Speech Understanding Report[Newell, etal., 1971, appendix 10].

Several papers from the field of programming languages and formal grammar theory discuss the effects of context; they have limited applicability to speech recognition systems, however. A summary of the methods is found in "Translator Writing Systems"[Feldman and Gries, 1968].

1.3. Outline of the Dissertation Presentation

Briefly, the plan of research is to investigate, in order: phonetic ambiguity, word ambiguity, lexical ambiguity, syntactic constraint and the combined effects of lexical ambiguity and syntactic constraint. A short preview of each chapter follows.

In chapter 2 we consider the major source of ambiguity; i.e., the acoustic speech signal itself. Several measures for quantifying phonetic ambiguity are investigated and compared. These measures provide a basis for the computation of lexical [illegible] rasal ambiguity in succeeding chapters.

In chapter 3 we present a model for lexical ambiguity. The model utilizes the knowledge of phone-to-phone confusions from chapter 2 and a general representation of the vocabulary to estimate the probability that an acoustic realization of some sequence of idealized phonemes will result in incorrect recognizion. The average number of expected words retrieved in an syntactically unconstrained lexical search is computed from these probabilities. This number is called the equivalent size of the vocabulary. The 10 digits have an equivalent size of 1.19 words, while the equivalent size of the spoken alphabet ("a", "b", ..., "z") is 3.87. This shows that the phonetic similarity of the alphabet is greater relative to the digits. This result is not surprising; it is, in fact, what one would expect.

Chapter 4 discusses the effects of syntactic restriction without regard to the lexical ambiguity. The syntax of languages for speech understanding systems impose restrictions on the number of word pairs, triples, etc. which can occur in the language. These limitations can dramatically reduce the total size of the search space. The IBM "New Raleigh" language has a 250 word vocabulary and an average sentence length of 8 words. Syntactic restrictions reduce the branching factor to 7.3. That is, on the average, one must disambiguate between 7 words. The voice programming language used by Lowerre has only a 37 word vocabulary and an average branching factor of 10.8. Thus, a 37 word vocabulary may provide a more stringent test of a recognition system than a 250 word vocabulary. This would depend also, of course, on the ambiguity of the words themselves.

Chapter 5 examines the combined effects of vocabulary ambiguity and syntactic complexity, but ignoring juncture ambiguity that further complicates connected speech (this can be thought of as a "best" behavior model or as a model for pause separated speech). This model assumes that word boundaries are known and therefore the only confusions that may arise are when two phonetically similar words have the same contexts. The effective branching factor obtained can be viewed as an optimistic representation of the expected behavior of the system.

The problems of connected speech are addressed in chapter 6. Given the "best" behavior model for complexity of chapter 5, we examine the limitations of that model with respect to the problems of connected speech. Then, a general model for ambiguity analysis of connected speech is developed. This model measures the ambiguity assuming that there is some uncertainty about the correctness of the recognition. In a sense, this may be viewed as a "worst" case model. The effective branching factor obtained is a pessimistic measure of the ambiguities which may arise.

Chapter 7 contains the analysis of four vocabularies and several languages of interest. The vocabularies are a set of 31 phones, the 10 digits, the spoken alphabet, and the alphabet and digits combined. The languages are CHESS, VP, LIZARD, IBM, LLBAS and LLEXT. CHESS is the original Hearsay-I chess task language. VP is a voice programming language with 37 words and LIZARD is a small version of VP having 17 words. IBM is IBM's "New Raleigh" language of english-like sentences. LLBAS is Lincoln Lab's "basic" language for displaying and controlling acoustic data. It has a vocabulary of 236 words. And LLEXT is an "extended" version of LLBAS having a 410 word vocabulary. Appendix C contains descriptions of these vocabularies and tasks.

There are many ways of approaching the analysis of ambiguity in speech

understanding tasks. Each new idea spawns several new and interesting problems and ideas. The methods we have used have been shown to be reliable relative estimators of ambiguity, although no claim is made that they are unique or complete. This work represents the best analytical tool we have to date for the design of languages for man-machine communication. These issues are discussed as part of chapter 8 on conclusions of this research.

2. PHONETIC AMBIGUITY

The major source of ambiguity in speech recognition is in the acoustic signal itself. Ambiguities of this nature must be dealt with at all levels of recognition. This chapter discusses the ambiguity of acoustic events and investigates several measures for quantifying its effects. These measures provide a basis for the computation of lexical and phrasal ambiguity in succeeding chapters.

Vocal production is accomplished by actions of the articulatory mechanism consisting of the lungs, vocal chords, tongue, lips and throat, mouth and nasal cavities. While the articulators can assume a wide variety of positions, only a few classes are employed by any one language. Each separately distinguishable class represents the same linguistic unit, called a phoneme. The acoustic realization of a phoneme is termed a phone. These realizations, unfortunately, do not fall into separable, mutually exclusive classes. The ambiguity of phones is well documented in experiments in both human perception and machine recognition. Some confusion exists in human perception with high quality speech when the phones are presented in isolation[Miller and Nicely, 1955]. This confusion becomes greater when the signal is corrupted by noise. Ambiguity in machine recognition is summarized nicely in the ARPA Speech Understanding Report[Newell, 1971]. This report also discusses ways of dealing with ambiguity in speech understanding systems and provides a good general reference for the subject.

2.1. Phonetic Ambiguity Measures

Most speech recognition systems begin by segmenting some parametric

representation of the acoustic space followed by classification of the resulting segments. Classification attempts to assign a phoneme-like label, or labels, to each segment. This chapter is concerned with the measurement of the reliability of making these classifications. In particular, we wish to determine the probability of phone p1 being recognized as phone p2 for all pairs. Although these probabilities are mathematically well defined, they cannot be calculated; they must be measured. We will discuss three ways of estimating these conditional probabilities: actual counts, acoustic-parametric metrics and theoretical models.

We could obtain these probabilities from actual counts using some existing recognition system, be it man or machine. This is usuallly done by comparing the output of the classifier with an accurate hand segmentation and labelling. The result is the classical confusion matrix giving the frequencies of correct and incorrect classifications. Conditional probabilities can then be derived from these frequencies. This method suffers from the fact that large amounts of data are required to provide accurate estimates and rare confusions, in general, are not accounted for. Also, the statistics could easily be biased by the particular design of the system used to gather them. Careful selection of the data is necessary in order that all phones are represented in their typical contexts. In human perception data, contextual cues which could provide information helpful for recognition must be eliminated.

Another method of obtaining the probabilities would be by direct comparison of parametric representations of the phones. In this method, a prototype is chosen from the set of realizations for each phone. Distances between phone pairs are then used to estimate the conditional probabilities. This method is also dependent upon the original data and the choice of the prototype. It does, however, consider rare events since it assigns some probability to every possible confusion.

All speech understanding sy tems must deal with the uncertainty of phone-phoneme similarity. There are almost as many methods of doing this as there are systems. Clearly then, no particular method stands out as the best. The choice of which method to use for estimating phonetic ambiguity represents a design decision. Since we are interested in a model which makes relative comparisons, any metric which captures the essence of the similarity and dissimilarity of the phones will serve the purpose. Of course, the closer the metric models the true probabilities, the more precise the outcome of the model. For the purposes of this thesis, we have chosen the minimum residual metric used by Itakura[1975]. Itakura's recognition scheme uses this metric along with a dynamic programming algorithm for temporal matching of isolated words. His system is one of the better telephone speech recognition systems. The minimum residual metric matches spectral characteristics of an unknown time signal with stored reference patterns. Reference patterns are essentially linear prediction models of the phones. The result of the matching algorithm is the log of the probability that the unknown is a realization of the stored model. Estimates of the phone to phone conditional probabilities for all phone pairs are obtained by treating the reference patterns as the unknowns. Appendix A contains a description of the algorithm, a set of reference patterns for the phones used in our analysis, and the complete phone probability matrix.

Another method for obtaining these probabilities would be through the use of a theoretical model. A long term goal is to develop an articulatory position model for estimating confusion probabilities. In the next section, we present such a model. At the present time, the model is not accurate enough to be used and represents an area for future research.

2.2. Articulatory Model

An articulatory feature model was chosen as the basis for arriving at a theoretical quantitative measure for phonetic ambiguity. Articulator positions are easily understood and represent a natural way of discussing phonetic phenomena. The model may be divided into five phases: selection of the features used, definition of phones in terms of these features, computation of distances in the feature space, inversion of distances to obtain log probabilities and normalization. We will discuss each of these in order.

The articulatory features used are listed in Figure 2-1 in decreasing order of influence. The set of allowed values is given for each feature.

Having decided on the features to be used, each phone was then defined in terms of these features in a fairly natural way. For instance, the throat is open for all vowels, turbulent for fricatives and constricted for the other consonants. A complete list of the definitions of the phones in terms of their feature values is given in appendix B.

The next step is to quantify the difference of phones based upon their feature descriptions. This part of the model assumes that the contributions of the articulators are essentially independent. Studies in co-articulation have shown that the movements of the articulators are not independent; and, later we will find that our model does incorporate one co-articulatory aspect. But, while co-articulation occurs often, its effects are minor. Thus, it was felt that the independency assumption retains sufficient information for our purposes. Furthermore, while these secondary effects may alter

1. Vocal Tract Closure	O- open
	C- closed or constricted
	T- turbulent
2. Vocal Chords	V- vibrating (voiced)
	U- not vibrating (unvoiced)
3. Nasal Cavity	O- open
	C- closed
4. Tongue Position	B- back
	C- central
	F- front
5. Tongue Height	L- low
	M- medial
	H- high
6. Tongue Tip	M- moving
	N- not moving
7. Lips	N- normal
	C- closed
	R- rounded

Figure 2-1. Articulatory Model - Features and Allowed Values.

absolute judgements, their effects will be partially nullified when making relative judgements using the same model.

The nature of the articulators and their features is such that the first two are very strong indicators of difference while the others are valid only when vocal tract closure characteristics and vocal chord vibration are the same. The decision part of the method is begining to emerge; if the first two features of the phones are the same, compute the ambiguity based upon the other features; otherwise, base the computation on the first two features alone. This decision process neglects one important consideration. When the velum, or soft palate, is opened, the combined nasal and mouth cavity presents a significantly different impedence for the driving function produced by the vocal chords. This co-articulation effect was incorporated into the model by splitting the voiced feature for the vocal chords into V for voiced and non-nasalized and N for voiced and nasalized. However, for purposes of the decision process described above, V and N are considered equal. The consequence of this modification will become clearer in the discussion of influence coefficients in the next few paragraphs.

Using our assumption of independency, each articulator may be assigned "influence coefficients" independently. These coefficients quantify the differences in the feature values. There will be one coefficient for each difference of feature values. Thus, each articulator will have either one or three coefficients depending upon whether it has two or three feature values. For example, one coefficient for vocal tract closure will be C(o,c)=C(c,o) representing the influence of the difference between the throat being open and the throat being constricted. Other coefficients for closure would be C(o,t)=C(t,o) and C(c,t)=C(t,c) representing the other possible ways closure

may differ. This gives a total of 17 coefficients. They are also given in appendix B. These coefficients were arrived at in an ad hoc manner by picking some starting values and modifying them until the response of the model seemed reasonable.

The complete flow chart for the computation is shown in Figure 2-2. The last box is a transformation from the distances computed into a space of log probabilities ranging from 0 to -2.0.

2.3. Validation of the Model

To test the soundness of the theoretical phonetic ambiguity model, the log probabilities from the theoretical model were correlated with probabilities derived from the Itakura metric. The results of this correlation are not at all encouraging. It is not sufficiently accurate to be of use at the present time. We hope to improve the model over the next few years.

Given the insufficiency of current theoretical models and the problems associated with perceptual data, it appears that the most convenient and accurate estimators of phonetic ambiguity are the acoustic-parametric metrics.

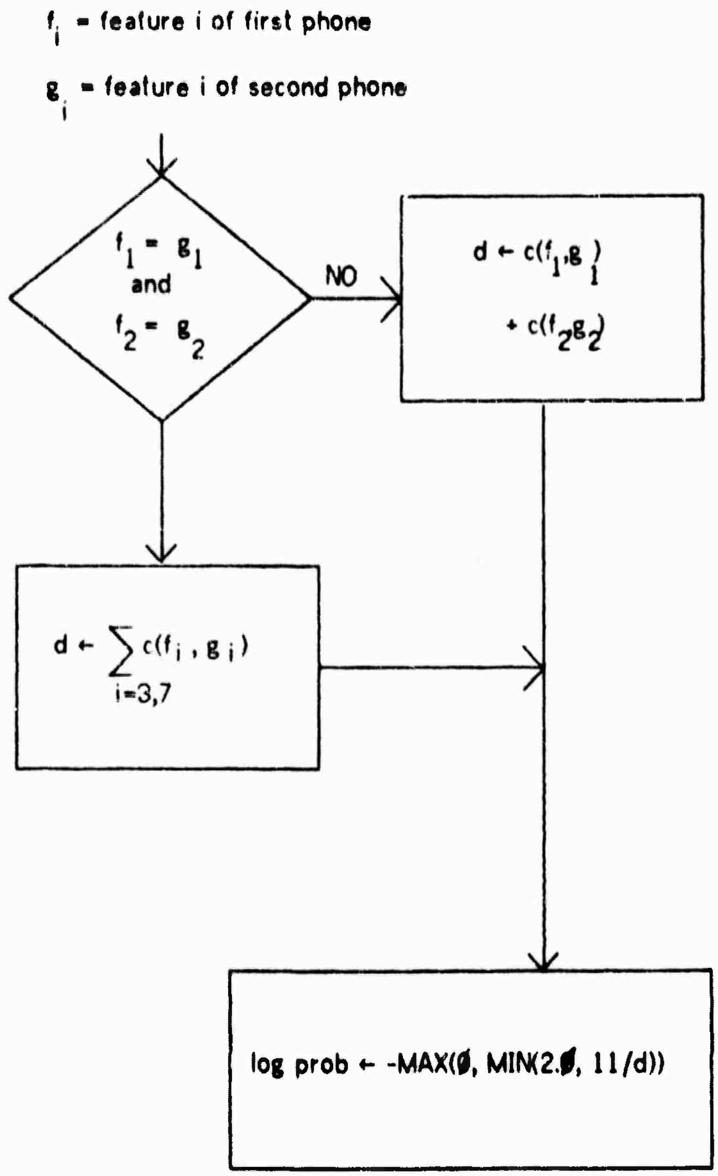


Figure 2-2. Flow Chart for Theoretical Phonetic Ambiguity Model.

3. LEXICAL AMBIGUITY

In this chapter we present a model for lexical ambiguity. The model utilizes the knowledge of phone-to-phone confusions from chapter 2 and a general representation of the vocabulary to estimate the probability that an acoustic realization of some sequence of idealized phonemes will result in incorrect recognizion. The average expected number of words retrieved in an syntactically unconstrained lexical search is computed from these probabilities.

3.1. The Nature of Lexical Ambiguity

Lexical ambiguity occurs when some word of the vocabulary (lexicon) is confused with another word because the two are phonetically similar. Thus, "six" and"sticks", being phonetically similar, could cause a lexical ambiguity if both exist in the same lexicon. Syntax may be useful in resolving this ambiguity. Syntactic restrictions will be covered in later chapters. This chapter will discuss the combinatorial explosion expected in pure bottom-up approaches as a result of lexical ambiguity.

How can two vocabularies with differing phonetic similarities be compared? Intuition may be reasonable for small vocabularies. Consider, for example the two vocabularies:

V1: "a", "b" & "c"
and V2: "zero", "nine" & "seventeen"

But is intuition good for larger vocabularies? Does intuition help in comparing a vocabulary of the 10 decimal digits and the 26 letters of the English alphabet with a vocabulary of 250 Japanese place names? These two vocabularies have been

recognized by the same system[Itakura, 1975]. So we have some basis for comparison. In this case the alphabet and digits were recognized with 88.6% accuracy and the place names with 97.3% accuracy.

The problem is to find a measure of the complexity of a vocabulary so that two may be compared. Briefly, the approach is to view the recognition process as a noisy channel and compute the information loss of the system. Information lost is a natural measure of the ambiguity, or complexity, of the system.

3.2. A Lexical Ambiguity Measure

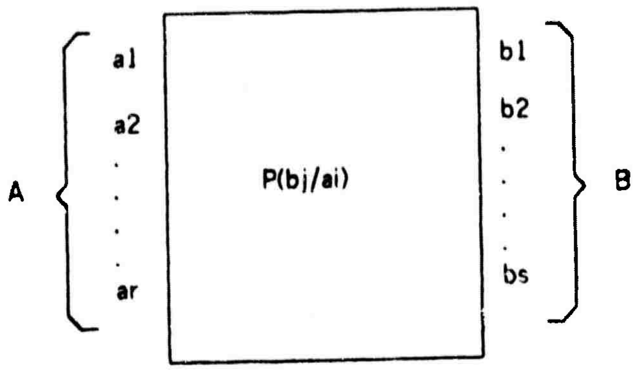
Figure 3-1 shows the block diagram of an information channel. There are r possible input symbols which may be chosen from alphabet A and s possible output symbols from the alphabet B. A channel is completely described by its channel matrix. This matrix consists of the set of conditional probabilities $P_{ij}=P(b_j/a_i)$ for all i and j, where P_{ij} is the probability that output symbol b_j is recognized when the input symbol a_i was spoken. In the context of word recognition, r=s, the input symbol represents the word spoken and the output symbol is the word recognized. An example of a channel matrix for the first three spoken letters of the alphabet is shown below.

$$\begin{array}{cc} & \begin{array}{ccc} \text{"A"} & \text{"B"} & \text{"C"} \end{array} \\ \begin{array}{c} \text{"A"} \\ \text{"B"} \\ \text{"C"} \end{array} & \begin{bmatrix} .992 & .007 & .001 \\ .007 & .971 & .022 \\ .003 & .090 & .907 \end{bmatrix} \end{array}$$

There are several important relationships among these probabilities. If some word a_i is spoken, then there is always some output. Thus,

$$\Sigma_A P(b_j/a_i) = 1 \quad i=1,2,\ldots r$$

Let the input symbols be chosen according to the probabilities $P(a_1)$, $P(_2)$, . . .



$$
\begin{array}{c|cccccc|}
 & b1 & b2 & b3 & \cdots & bs \\
\hline
a1 & P11 & P12 & P13 & \cdots & P1s \\
a2 & P21 & P22 & P23 & \cdots & P2s \\
a3 & P31 & P32 & P33 & \cdots & P3s \\
 & \cdot & & & & \\
 & \cdot & & & & \\
as & Pr1 & Pr2 & Pr3 & \cdots & Prs \\
\hline
\end{array}
$$

Figure 3-1. An Information Channel and its Channel Matrix.

$P(a_r)$. These are refered to as the *a priori* probabilities of the input symbols. Then the output symbols will appear according to some set of probabilities $P(b_1)$, $P(b_2)$, . . . $P(b_r)$. The dependency between these two distributions is given by

$$P(b_j) = \Sigma_A \; P(a_i)\, P(b_j/a_i)$$

The probabilities $P(b_j/a_i)$ used to describe a channel are called the forward probabilities. The backward probabilities $P(a_i/b_j)$ may derived using Bayes' Law as

$$P(a_i/b_j) = P(a_i,b_j)*P(a_i) = \frac{P(b_j/a_i)*P(a_i)}{P(b_j)}$$

Where $P(a_i,b_j)$ is the probability of the joint event (a_i,b_j). These $P(a_i/b_j)$ are also called the *a posteriori* conditional probabilities of the input symbols.

We will next discuss information quantities relating to the channel model. The information received when a_i is spoken and b_j is recognized is[Goldman, 1953]

$$I(a_i;b_j) = \log \left[\frac{\textit{a posteriori}\text{ probability that } a_i \text{ was spoken given that } b_j \text{ was recognized}}{\textit{a priori}\text{ probability that } a_i \text{ was spoken}} \right]$$

$$= \log[\; P(a_i/b_j)/P(a_i) \;]$$

The exponent for the log function is arbitrary and defines the information units. An exponent of 2 will be used throughout this thesis. Thus, information is measured in bits. If the channel is perfect, then $P(a_i/b_i) = 1$ for all i, and the information per message is

$$H(a_i) = -\log[\; P(a_i) \;]$$

The average information per message is the average $I(a_i;b_j)$ over all events (a_i,b_j).

$$H(A) = -\Sigma_{A,B}\ P(a_i,b_j) \log [\ P(a_i)\ ]$$
$$= -\Sigma_A\ P(a_i) \log P[\ P(a_i)\ ]$$

This quantity is the average information transmitted. It is also called the ***a priori*** uncertainty of the input alphabet. Note that it depends only on the ***a priori*** probabilities. If each input symbol is equally probable, then $P(a)=1/r$ and

$H(A) = \log r$ bits/symbol

$H(A)$ is the average number of bits necessary to specify a symbol of the alphabet.

The average information received at the output of an imperfect channel is

$$I(A;B) = \Sigma_{A,B}\ P(a_i,b_j)\ I(a_i/b_j)$$
$$= \Sigma_{A,B}\ P(a_i,b_j) \log \left[\frac{P(a_i/b_j)}{P(a_i)} \right]$$
$$= -\Sigma_{A,B}\ P(a_i) \log [\ P(a_i)\ ] + \Sigma_{A,B}\ P(a_i,b_j) \log [\ P(a_i/b_j)\ ]$$
$$= -\Sigma_A\ P(a_i) \log [\ P(a_i)\ ] + \Sigma_{A,B}\ P(a_i,b_j) \log [\ P(a_i/b_j)\ ]$$
$$= H(A) + \Sigma_{A,B}\ P(a_i,b_j) \log [\ P(a_i/b_j)\ ]$$

Rewriting

$$H(A) - I(A;B) = -\ \Sigma_{A,B}\ P(a_i,b_j) \log [\ P(a_i/b_j)\ ]$$

Written this way, we see that the right hand side is equal to the information transmitted minus the information received. This quantity, call the ***equivocation*** and denoted $H(A/B)$, represents the information lost in the channel.

$$H(A/B) = \Sigma_{A,B}\ P(a_i/b_j)\ P(b_j) \log[\ P(a_i/b_j)\ ]$$
$$= -\ \Sigma_B\ P(b_j)\ \Sigma_A\ P(a_i/b_j) \log[\ P(a_i/b_j)\ ] \qquad (3\text{-}1)$$

$H(A/B)$ is the average number of bits necessary to specify an input symbol after examining the output. Recalling that $2^{H(A)}$ measures the actual size of the vocabulary, consider $2^{H(A/B)}$. This quantity, which we call the equivalent vocabulary size, or EVS, is a measure of the size of the vocabulary given the loss due to ambiguity in the vocabulary.

For perfect recognition , $H(A/B)=0$ and the EVS is 1 word. This occurs when $P_{ii}=1$ and $P_{ij}=0$ for $i \neq j$; stated another way, when every word is phonetically unambiguous. At the opposite extreme, every word is phonetically identical to every other word. Then $H(A/B)=H(A)$ and the information received is 0. In this case, $2^{H(A)}$ bits are required to represent an input symbol after examining the output. If each symbol is equally probable the interpretation is that the best one could do would be to make a guess from among the r possible words.

Only the probabilities $P(a_i/b_j)$ and required to calculate the EVS of a vocabulary. We will now discuss how these may be obtained.

3.3. Word Ambiguity Model

The natural method for obtaining the conditional probabilities would be to take actual counts using some existing system. The same problems exist here as for the phone-phone probabilities in chapter 2. To repeat, they require large amounts of carefully selected data for accurate estimates and the data will be biased by the idiosyncrasies of the system used to gather the data. There are methods of obtaining this data which are more feasible. We have investigated three methods.

M1: matches network representations against other network representations using Itakura's metric for log probabilities.

M2: matching network representations against acoustic realizations using Harpy with the Itakura metric[Lowerre, 1976].

M3: matching acoustic realizations against acoustic realizations using Itakuras recognition scheme[Itakura, 1975].

The last two methods are recognition systems which result in a set of conditional probabilities $P(a_i/s_j)$ for words a_i given acoustic signal s_j.

The first method requires a model for matching network representations. The model chosen is general and is as independent of any particular recognition scheme as possible. It performs worst case analysis in that it finds the match which maximizes the probability of confusion. The next sections discusses this model in more detail.

The phonetic definition of each word in the vocabulary is embodied in a finite state recognition network similar to the networks used by HARPY[Lowerre, 1976]. An example of a recognition network for "A" and "B" is shown in Figure 3-2. The network contains an initial state S_0 and a final state S_f. Every allowed variation of a word of the vocabulary is represented by a subnetwork starting at S_0 and ending at S_f. Each subnetwork is buffered at the beginning and end by an optional silence phone ("-") so that initial and final stops and fricatives have a context which they may match. Each state of the network contains the phone label representing that state. Let this be called PHNOF(S); for instance, PHNOF(S_2)=EH. In addition, there is a word associated to

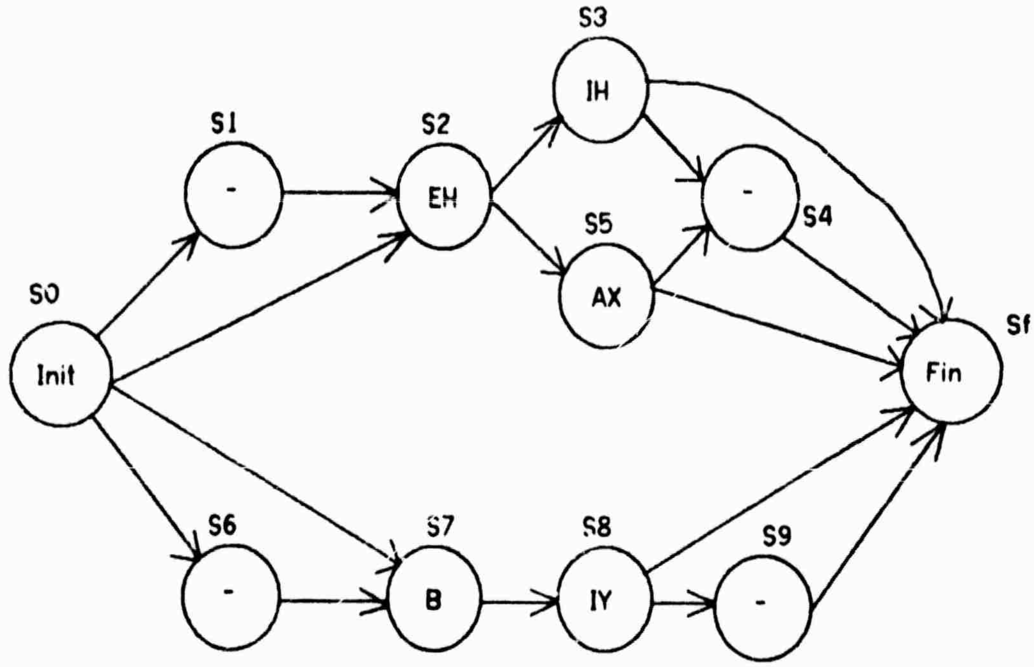


Figure 3-2. Word Network Example, "A" and "B".

every state. This has been omitted for clarity. Let this correspondence be given by WORDOF(State). Thus, WORDOF(S_2)="A". For each state, the set of immediately previous states is denoted PREV(S). For example, PREV(S_2)={S_0,S_1}. The set f(W) is all final states of word W:

$$f(W) = \{s \mid s \in PREV(S_f) \text{ and } WORDOF(s) = W\}$$

For word "A", f("A") = {S_3,S_4,S_6}.

From such a word network and the phone-phone probabilities, word-to-word confusion probabilities are calculated. This is done by first computing state-to-state confusion probabilities $P(S_i/S_j)$. Then, word-to-word probabilities are extracted using

$$P(W_i/W_j) = \max_{\substack{S_i \in f(W_1) \\ S_j \in f(W_2)}} P(S_i/S_j)$$

Since these relative probabilities are maximized, they do not in general sum to one. They must be normalized so that

$$\Sigma_{j=1,r}\, P(W_i/W_j) = 1.0 \qquad i=1,2,\ldots r$$

The effective vocabulary size defined in the previous section is then computed from this matrix using equation 3-1 (page 21). This brief description serves as a guide to the discussion of the next section.

The flow diagram for the computation of state confusion probabilities is shown in figure 3-3. In this algorithm, all probabilities have been replaced with their logs so that multiplications become additions. For each word W, the probabilities $P(W_i/W)$ are found. Given a word W, a partial order exists for the states in its subnetwork. For example, the partial order for "A" is (S_0,S_1,S_2,S_3,S_5,S_4,S_f). This partial order determines the order in which the calculations proceed. First, $P(S_0/S_0)$ is set to

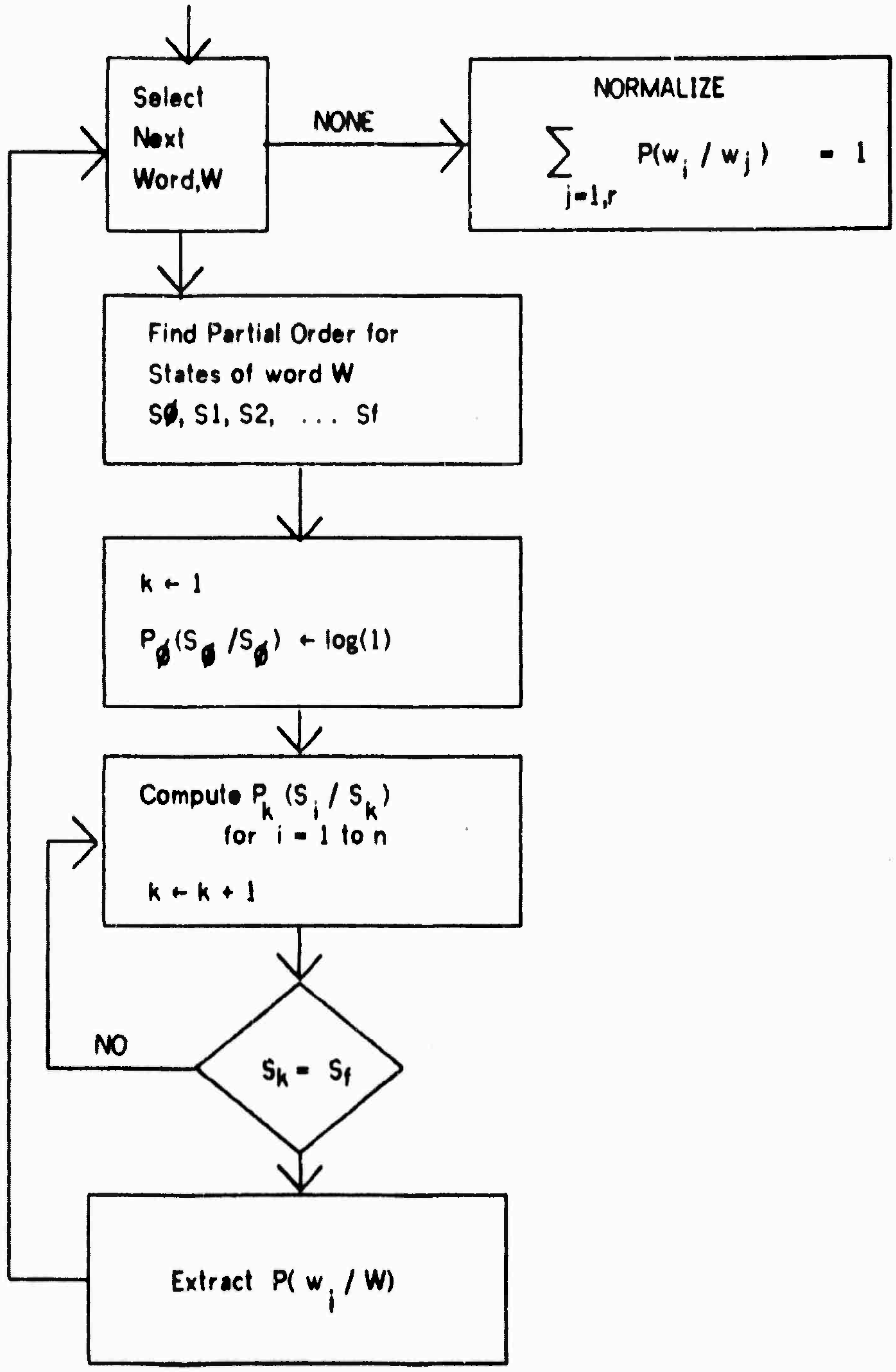


Figure 3-3. Flow Diagram for Word to Word Probability Calculation.

log(1)=0. This may be interpreted as: the probability of being in state S_0 given that you should be in state S_0 is 1. The computation then proceeds using the recursive formula:

$$P_k(S_i/S_k) = \max_{\substack{Q' \in \{PREV(S_i) \cup S_i\} \\ Q \in PREV(S_k)}} P_{k-1}(Q'/Q) + PHNPRB[PHNOF(Q), PHNOF(Q')]$$

The subscripts on P are redundant, but serve to emphasize that the probabilities on each side of the equation are separate quantities. Figure 3-4 helps to interpret this equation as follows: The first term on the right represents the maximum of the probabilities of being in previous states of S_i given that the correct state should have been some previous state of S_k. Added to this is the (log) probability of misrecognizing the acoustics as PHNOF(Q') given that PHNOF(Q) was spoken. The result is the probability of being in state S_i given that the correct recognition would lead to state S_k. Allowing Q' to be S_i serves two purposes. First, sequences of phones may match a single phone. For example, consonantal clusters may match a single consonant. Or, as in the example shown, the diphthong <EH IH> to match the vowel <IY>. Secondly, it may happen that the best match occurs before $S_k=S_f$. This would be true when W ended with a stop consonant which matched the optional silence of another word. Since from then on, PHNPRB[PHNOF(),PHNOF()]=0, the self cycling nature of the definition will retain the maximum match until $S_k=S_f$.

3.4. Interpretation of Results

Figure 3-5 lists the results of lexical analysis for several vocabularies. Recall that an equivalent vocabulary size of 1 indicates no information is lost in the channel and thus recognition is perfect. The first three vocabularies, ABC, BDV and V123 are the vocabularies introduced as intuitive exercises in chapter 1. We see that BDV is

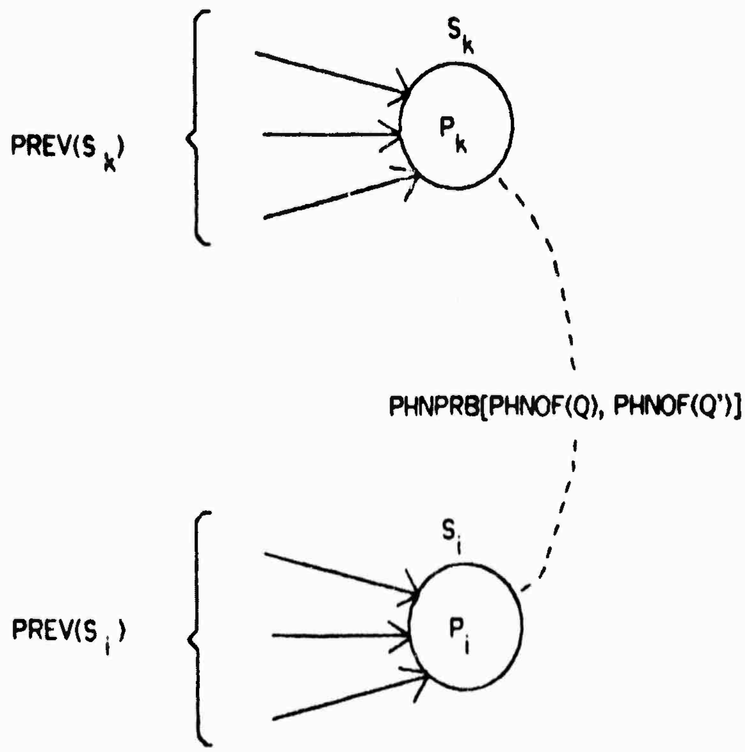


Figure 3-4. Calculation of $P(S_i / S_k)$.

Task	Number of Words in Vocabulary	Equivalent Vocabulary Size
ABC	3	1.19
BDV	3	1.99
V123	3	1.03
PHONES	33	20.10
DIGITS	10	1.19
ALPHABET	26	3.87
ALPHA-DIG	36	3.41
CHESS	25	1.46
Lincoln Labs		
Basic	236	2.43
Extended	410	3.54
IBM	250	2.31
LIZARD	17	1.55
VP	37	1.70

Figure 3-5. Results of Lexical Ambiguity Analysis.

obviously the most ambiguous of the three. The fact that ABC and V123 are so close in difficulty may be a slight surprise. The phones are highly ambiguous, as expected. The 10 digits have an equivalent vocabulary size of 1.19 words while the equivalent size of the spoken alphabet is 3.87 words.

Consider now, the other vocabularies. Their real sizes range form 17 to 410 words while the effective sizes range between 1.46 and 3.54 They seem to be directly related. This relative order is expected since large vocabularies have greater potential for ambiguity and therefore, in general, have larger effective sizes. An interesting comparison can be made between the chess vocabulary and the Lizard vocabulary. In this case, the 17 words of the Lizard vocabulary have slightly higher ambiguity than the 25 words of the chess task.

4. SYNTACTIC RESTRICTION IN SPEECH UNDERSTANDING TASKS

The syntax of languages for speech understanding systems impose restrictions on the number of word pairs, triples, etc. which can occur in the language. These limitations can dramatically reduce the total size of the search space. This chapter discusses the effects of syntactic restriction without regard to the similarity of the words involved. The combined effects of vocabulary and syntax are examined in chapters 5 and 6.

4.1. Measures of Grammatical Complexity

Some measures of grammar size which have been used are the the number of non-terminals and the number of productions(right hand sides) in the grammar. There are, in general, many ways do define a particular language. Thus, these are only very gross measures in that they represent the complexity of the representation of the grammar as opposed to the complexity of the grammar or syntax itself. Better measures for quantifying complexity are the number of pairs of words that may occur together and the number of word triples that may occur in language. Pairs and triples give some idea how syntax restricts the search space, but fall short in two aspects. First, they account for local context only; that is they consider at most the preceding and following words. Secondly, they say nothing about the probabilities with which they occur. In this chapter, average branching factor will be discussed as a measure of syntactic restriction. Average branching factor(ABF) is defined as the expected number of words which may occur next in an utterance. Two methods of averaging will be presented, resulting in two types of ABF. Static average branching factor is the result of averaging uniformly over all possible states of recognition. Dynamic

branching factor is computed similarily, but includes the probabilities of being in the states. Thus, states which are rarely visited do not contribute as much as those which occur often, such as those that occur in every sentence. While computing the average branching factor, maximum and minimum branching factors are also found. For completeness and comparison, all quantities mentioned above are tabulated for the languages investigated. Fundamental to the computations is the method of representing the syntax.

The initial representation for a grammar is its Backus Normal Form or Backus-Naur Form(BNF) definition. An example of a BNF is shown in Figure 4-1 for the very simple task called APEX. This example will be used to illustrate the concepts presented this chapter. This task is not typical(see appendix C), but is purposefully small so that important ideas may be presented clearly. This BNF is transformed into a probabilistic grammar network. Recognition networks of this form have been studied by several investigators [Fu 1969, Woods 1970, Baker 1975, Lowerre 1976]. Recreating previous work at this level was deemed unnecessary and unjustified. Thus, we chose to utilize the network representation used by the Dragon speech recognition system[Baker,1975] and later modified for use by the HARPY system[Lowerre,1976]. The network for APEX is shown in figure 4-2. In this figure, each box represents a state of partial recognition and is labelled with a state number. There is a special state called the initial state, denoted here by S_0. Every other state contains a word from the vocabulary. The successors for each state are indicated by arrows in the figure. The set of successors for state S_k is denoted by NEXT(S_k). If NEXT(s) is empty, the state s is called a final state. In similar fashion, the set of predecessors is represented by the function PREV(s). By "being in a state" we mean that some partial recognition has led to the state after recognition of the word of the state. Loops are

```
<QUERY>::=               [ <REQUEST> ]

<REQUEST>::=             HELLO
                         GIVE <GIVE>

<GIVE>::=                MORE
                         EVERYTHING
                         ME <NOUN-PHRASE>

<NOUN-PHRASE>::=         EVERYTHING
                         THE <NOUN>

<NOUN>::=                NEWS
                         SUMMARY
                         STORIES
```

Figure 4-1. BNF definition for the example APEX.

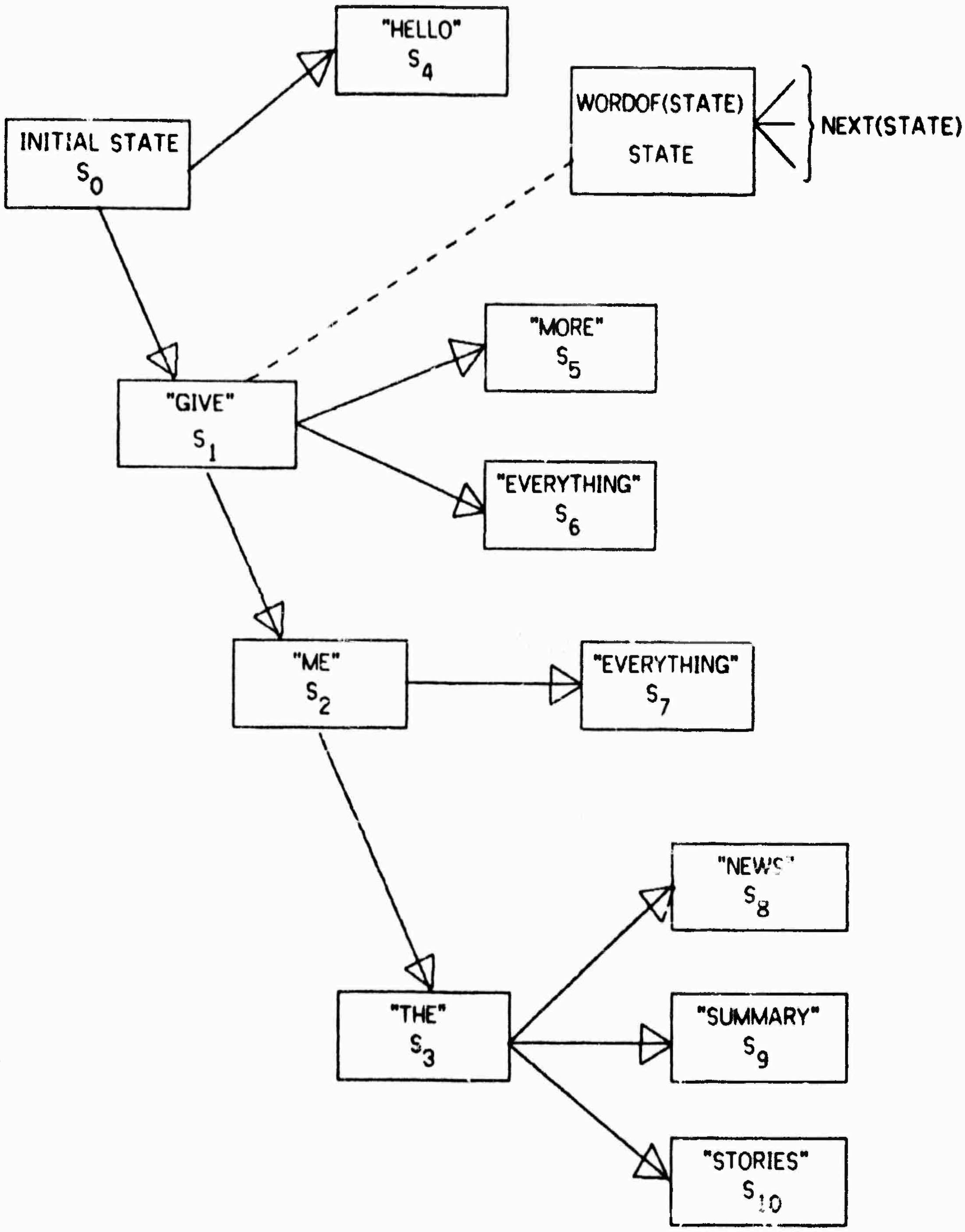


Figure 4-2. Example of a Grammar Network.

possible in this representation, although they do not occur in the specific example. It is clear that any regular(finite state) language can be represented by such a network. Since most speech recognition tasks have been defined in terms of regular languages, this does not represent a severe restriction. Furthermore, languages for speech recognition are designed to describe a large but finite number of sentences; it is an artifact that they also define sentences of infinite length. Thus, one could, in general, redefine the language by describing all the sentences of interest using finite-state grammars. In fact, very simple transformations allow this to be done. The Lizard task[appendix C], for example, contains phrases which could be defined by

<SIMPLE-EXPRE>::=<PRIMARY> <BIN-OPE> <PRIMARY>

By defining the non-terminals <PRIMARYCE> and <PRIMARYDE>, each with identical but separate definitions, this may be rewritien as

<SIMPLE-EXPRE>::=<PRIMARYCE> <BIN-OPE> <PRIMARYDE>

This transformation has preserved context by duplicating a non-terminal which occurred in different contexts. Each of the tasks investigated was found capable of being made regular by this method.

Given the BNF a grammar network, the simple measures can be found quickly. The number of non-terminals and productions is determined by counting these quantities directly from the BNF. All possible word pairs (and triples) may be obtained by considering each state in the network and its possible successors (and predecessors). A complete list of the word sequences for the example APEX is given in figure 4-3. These four simple measures are summarized for all the tasks in figure 4-4. While these quantities are useful, a more revealing quantity is the average branching factor.

Number of Word Pairs 16		Number of Word Triples 15		
GIVE	MORE	GIVE	ME	EVERYTHING
GIVE	EVERYTHING	GIVE	ME	THE
GIVE	ME	ME	THE	NEWS
ME	EVERYTHING	ME	THE	SUMMARY
ME	THE	ME	THE	STORIES
THE	NEWS	GIVE	MORE	#
THE	SUMMARY	GIVE	EVERYTHING	#
THE	STORIES	ME	EVERYTHING	#
HELLO	#	THE	NEWS	#
MORE	#	THE	SUMMARY	#
EVERYTHING	#	THE	STORIES	#
NEWS	#	#	GIVE	EVERYTHING
SUMMARY	#	#	GIVE	ME
STORIES	#	#	GIVE	MORE
#	GIVE	#	HELLO	#
#	HELLO			

Figure 4-3. Word Sequences for the Example APEX.

TASK	NNT	NPS	PAIRS	P/WORD	TRIPLES	T/WORD
CHESS	33	84	207	7.96	2362	94.48
LIZ	8	34	182	10.71	1866	109.76
VP	41	181	622	16.81	10,152	276.37
IBM	38	314	2304	9.22	22,004	88.02
LLBAS	127	391	2617	11.09	47,219	200.08
LLEXT	163	679	10,286	25.03	566,633	1382.03

NNT is the number of Non-terminals.
NPS is the number of Productions.
PAIRS is the number word pairs.
P/WORD is the number of word pairs/word.
TRIPLES is the number of word triples.
T/WORD is the number of word triples/word.

Figure 4-4. Some simple measures of grammatical complexity.

Two methods of averaging are defined yielding a static average branching factor (SABF) and a dynamic average branching factor(DABF). Let BR(s) be the local branching factor for the state s. BR(s) is the number of states in NEXT(s). SABF is BR(s) averaged over all non-final states. Define

$$NFS = \{ s \mid NEXTX(s) \text{ is not EMPTY} \}$$

Then

$$SABF = \frac{\Sigma_{s \in NFS} BR(s)}{| NFS |}$$

While finding this average, maximum and minimum branching factors are also found. The result of this calculation for the example of this chapter is

Average Branching Factor = 2.5
Maximum Branching Factor = 3.0
Minimum Branching Factor = 2.0

4.2. Dynamic Branching Factor - A Measure of Syntactic Restriction

The static method of averaging does not account for the fact that the sentence "Hello" may occur fewer times than sentences described by the other paths. This may be done by assigning transition probabilities to each arc in the network. These probabilities represent the relative frequencies of the alternative paths at each state. The transition probabilities on the arcs leading from each state, say s, to the set of next states sum to one.

Figure 4-5 shows the APEX network with these transition probabilities placed on the arcs. Let P(s/r) be the probability of going to state s given current state r. From these transition probabilities we calculate P(s/t), the probability of being in state s at

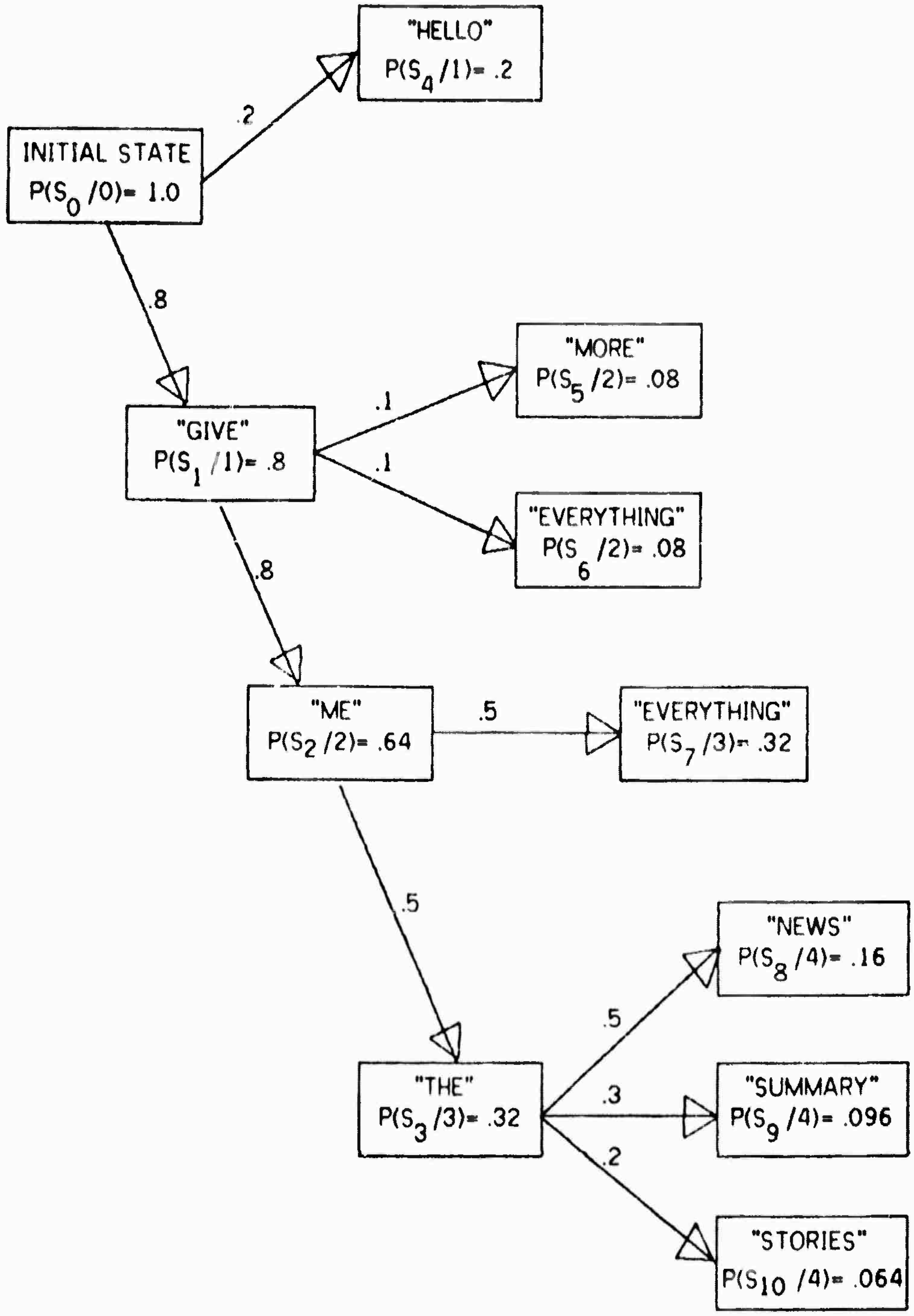


Figure 4-5. State Probabilities for the example APEX.

time t. Time is measured in words in this case. These probabilities are defined recursively by

Assign: $P(s_0/0)=1$

Define: $P(s/t) = \Sigma_{r \in PREV(s)} R(s/r)\, P(r/t-1)$

Figure 4-5 shows these probabilities for all states and times which result in non-zero probabilities.

Average sentence length(ASL) is a simple sum of these state/time probabilities over all non-final states and time.

$$ASL = \Sigma_T \; \Sigma_{s \in NFS} \; P(s/t)$$

Thus, the average sentence length for this example is 1.0+.8+.64+.32= 2.76 words/sentence.

Dynamic branching factor may now be defined as follows. First, find the sums of the log of the local branching factors probabilistically weighted and averaged over all time. That is,

$$LWS = \frac{\Sigma_T \; \Sigma_{s \in NFS} \; P(s/t) \log [\, BR(s) \,]}{\Sigma_T \; \Sigma_{s \in NFS} \; P(s/t)}$$

Note that the denominator in the above expression is simply the average sentence length. Dynamic branching factor is then 2 to the exponent LWS.

$$DABF = 2^{LWS}$$

Transition probabilities are necessary for the computation of dynamic branching factor and average sentence length. These probabilities vary depending on the users

preferences and the particular problem he is trying to solve in the task domain. Learning these probabilities is a current topic of research in speech recognition[Bahl et. al., 1976] For the purposes of computation, the transition probabilities have been chosen such that

$$P(s,r) = 1/K$$

$$\text{where} \quad K = \begin{cases} |\,NEXT(s)\,| & \text{if } s \in NFS \\ |\,NEXT(s)\,| + 2 & \text{otherwise} \end{cases}$$

The probability of moving to a new state is roughly uniform over all possible next states, with some preference to termination if the current state is a final state. This distribution assigns slightly higher weights to the shorter sentences.

The example presented in this chapter was not recursive and, therefore, the sum over time terminated properly. In the case of recursion, some stopping criteria is necessary. Since the computation time is not excessive in this calculation, a very loose constraint is used. The computation is stopped whenever all sentences of length 100 or less have been considered(t=100) or whenever the probability of remaining sentences falls below 0.00001, whichever comes first. In the tasks examined, the only task which went to sentences of length 100 was the Chess task. The residual probability of sentences of greater length was .000011 in this case.

Average branching factors and sentence lengths are summarized in Figure 4-6. This table contains the average, maximum and minimum static branching factors, the dynamic branching factor and the average sentence length. The dynamic branching factor from figure 4-6 assigns a relative ordering to the complexity of the tasks. That order is CHESS, IBM, LLBAS, LIZ, VP and LLEXT. Static branching factor yields the ordering CHESS, IBM, LIZ, LLBAS, VP and LLEXT. Maximum branching factor yields

TASK	STATIC BRANCHING FACTOR AVE	MAX	MIN	DYNAMIC BRANCHING FACTOR	AVERAGE SENTENCE LENGTH
CHESS	8.65	21	1	7.36	8.10
LIZ	10.78	11	6	9.32	6.08
VP	14.11	37	1	10.82	8.22
IBM	10.58	24	1	7.73	8.09
LLBAS	11.34	61	1	9.15	7.52
LLEXT	25.32	161	1	20.28	8.93

Figure 4-6. Branching factors for the Tasks Studied.

much the some order except for the Lizard task. Minimum branching factor gives little information since it is usually one. In all cases, the dynamic branching factor is lower than the static branching factor. This is true because the log function gives higher weights to small branching factors and, in general, the larger local branching factors are found in the longer, and therefore less probable, sentences. The Lizard task has the lowest average sentence length; 6.08 words per sentence. The average sentence length falls between 7.5 and 9 words for the other tasks. It is interesting that the chess task has the lowest branching factor and the one of the largest average sentence lengths. This means that individual decisions are easier, but there are more decisions to be made. This leads to the notion of search space.

4.3. Syntactic Search Space

The average branching factor described above is a local measure of complexity. It represents the degree of difficulty of making individual decisions. The total size of the search space is a global measure of the complexity. The syntactic search space is the size of a tree with this average branching factor and having depth equal to the average sentence length. Since this number would be quite large, it is more convenient to use the log of this quantity. Thus,

$$\log[\text{Search Space Size}] = \text{ASL} * \log[\text{ DABF }]$$
$$= \text{ASL} * \text{LWS}$$

The log of the search space size for each of the tasks under consideration is given in Figure 4-7. This number is, roughly, the number of binary decisions necessary to recognize a sentence (considering syntax only). The relative ordering is now LIZ, IBM, CHESS, LLBAS, VP and LLEXT. Lizard has moved down because there are fewer decisions, on the average. Chess has moved up high in the ranking because of its long

TASK	log[Search Space Size]
CHESS	23.31
LIZ	19.56
VP	28.24
IBM	23.23
LLBAS	24.01
LLEXT	38.79

Figure 4-7. Log of Search Space Size.

sentence length. In practice, the average sentence length for the chess task is on the order of 6 or 7 words. This is probably due to the "principle of least effort". In the Chess task, moves may be said in a variety of ways and people will usually opt for the smallest unambiguous sentence. Independent estimates of the average sentence length could be used for this calculation, if they were available.

5. COMPLEXITY IN CONNECTED SPEECH - A RESTRICTED MODEL

Chapter 4 discussed how syntax restricts the number of word combinations allowed in the language. Further restriction is possible when the syntax eliminates confusable words from appearing within the same context. This chapter examines the combined effects of vocabulary and syntax for connected speech in a restricted model. A general model for connected speech is presented in chapter 6.

The model used in this chapter assumes that the recognition process is "well behaved" in the sense that it proceeds almost entirely without error. That is, each word of the utterance is assumed to have been recognized correctly as the process moves from one correct state to another. The model therefore measures the average ambiguity encountered during a correct recognition. Another view is that this is a model for ambiguity in pause separated speech. We will refer to this as the "best" case model.

5.1. Lexical Ambiguity and Syntactic Restriction

In chapter 4 the calculation of dynamic branching factor used the log of the local branching factor as the quantity which was averaged. This may be interpreted to mean that local alternatives are viewed as a set of entirely confusable words. This is never true and, in fact, a well designed language will use the syntax to place acoustically similar words in different contexts. Figure 5-1 gives the BNF description of the Lizard task language. The word pair having highest acoustic similarity in this task is "ADD" - "EIGHT". Figure 5-2 shows the initial state of the Lizard grammar network along with its successors. Define the sub-vocabulary of a state s to be the

<UTT>::=	[<COMMAND>]
<COMMAND>::=	<OP><SIGN-NUMBER> DISPLAY
<OP>::=	ADD SUBTRACT MULTIPLY DIVIDE LOAD
<SIGN-NUMBER>::=	MINUS <NUMBER> <NUMBER>
<NUMBER>::=	<DIGIT> <DIGIT><NUMBER-2>
<DIGIT>::=	ZERO ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE
<NUMBER-2>::=	<DIGIT-2> <DIGIT-2><NUMBER>
<DIGIT-2>::=	ZERO ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE

Figure 5-1. BNF Description for the Lizard Task.

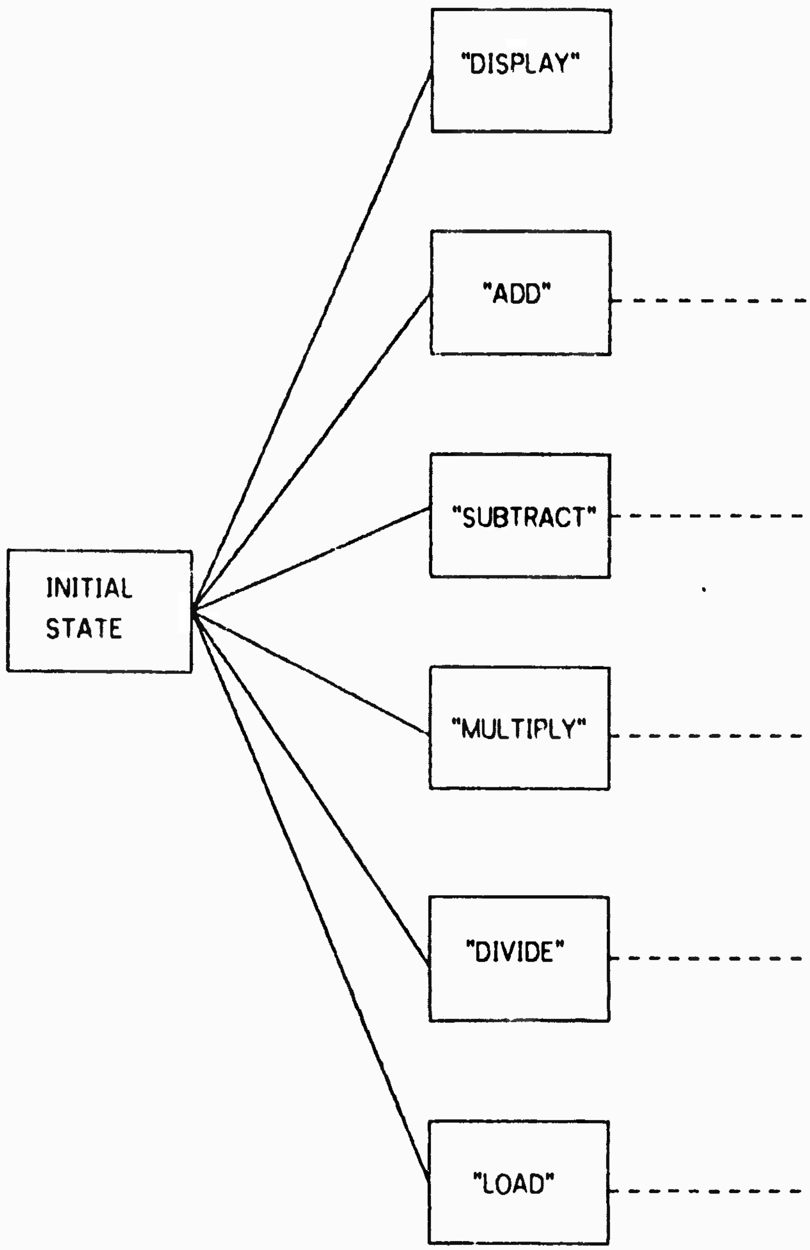


Figure 5-2. An Example of a Sub-vocabulary in the Lizard Task.

set of words determined by the successors of state s. The only sub-vocabulary containing the word "ADD" is the sub-vocabulary of the initial state. Note that it does not contain the word "EIGHT". The syntax has isolated these two words from one another in such a way that they would never cause an ambiguity, assuming that no errors have yet occurred. In this particular example, the only time these two words could be confused would be if the beginning word "ADD" was misrecognized as silence and the second word of the utterance was "EIGHT". This may happen if "ADD" were reduced or swallowed, the speech/no speech detector failed, or, more likely, the words were run together so that the <D> went undetected and the two vowels were missegmented as one vowel. If an error of this nature was made, then "EIGHT" could easily be misrecognized as "ADD". Such problems will be addressed in chapter 6.

5.2. Ambiguity Analysis in the Restricted Model

To combine the effects of vocabulary restriction and syntactic restriction, the branching factor is replaced with the effective branching factor(equivalent vocabulary size) for the sub-vocabularies of each state in the calculations performed in chapter 4. The effective branching factor for sub-vocabularies is computed in the same way equivalent vocabulary size was computed in chapter 3. Recall that the effective vocabulary size for the Lizard vocabulary was 1.55 words. In the example of figure 5-2, the local branching factor is 6 and the effective branching factor is 1.16.

The branching factors computed in chapters 3, 4 and this chapter are tabulated in figure 5-3. The first two columns of this table contain the task name and the number of words in the vocabulary of the task. The columns to the right contain branching factors under various conditions. The effective branching factor for the vocabulary, without the effects of syntactic restriction, is shown the column labeled

Task	Number of Words in Vocabulary	BRANCHING FACTORS: Vocabulary Only	Grammar Only	Vocabulary and Grammar
PHONES	33	20.10	33	20.10
DIGITS	10	1.19	10	1.19
ALPHABET	26	3.87	26	3.87
ALPHA-DIG	36	3.41	36	3.41
CHESS	25	1.46	7.36	1.09
Lincoln Labs				
Basic	236	2.43	9.15	1.20
Extended	410	3.54	20.28	1.34
IBM	250	2.31	7.32	1.09
LIZARD	17	1.55	9.32	1.46
VP	37	1.70	10.82	1.28
VPNS	37	1.70	37.00	1.70

Figure 5-3. Results of Complexity Analysis.

"vocabulary only". It is the same as the effective vocabulary size described in chapter 3 and represents the average number of words retrieved in a lexical match per word spoken. Thus, for the 10 digits, 1.19 words would appear, on the average, for each word spoken. The column marked "grammar only" gives the average branching factor considering syntax, but disregarding the effects of lexical ambiguity. This branching factor, described in chapter 4, represents the average fan-out of the syntax; or, the average number of words which may follow another word in an utterance. This column is the same as the vocabulary size for the first 4 tasks since any word may follow any other word. For the tasks with syntactic constraints, this branching factor ranges from 7.32 words to 20.28 words. The last column contains the effective branching factor considering the combined effects of lexical ambiguity and syntactic constraint.

We will first consider the tasks in order and then general aspects of the complete table. Recall that the phone task vocabulary was just the set of phones. The effective vocabulary size obtained is 20. This means that every phone, on the average, matches uniformly to 20 phonetic labels. It must be remembered that this is for isolated phones without syntactic support, or even a surrounding lexical context. Even so, this value seems rather high. This quantity has been computed from actual counts from the BBN speech recognition system [Makhoul, 1975]. The value for their system, which uses 67 different phoneme types and 83 acoustic classifications, is 4 labels/segment. If this figure were used as a standard, it says that the computation of H(A/B) is roughly two and one-half times larger than it should be. If anything, this implies that our model accounts for more variability in the phones than is really there; that is, it is biased away from high quality, well articulated speech. We intended this to be the nature of the system. Also, bear in mind that the models were designed for relative comparisons.

For the 10 digits, the effective vocabulary size is 1.19. The interpretation here is that six words will be retrieved for every five words spoken and one of them is obviously wrong. This corresponds, roughly, to a recognition rate of 83%. Currently, speech recognition systems have very little trouble recognizing the digits spoken in isolation. Again, we see that if the model is biased, it is biased toward greater variability. We feel that this is actually an advantage of the model; for, given the relative soundness of the model, the differences between vocabularies are enhanced.

The spoken alphabet exhibits an effective vocabulary size of 3.87 words. This is reasonable, particularly when compared to the digits, since the spoken alphabet is highly ambiguous.

In the alphabet-digit vocabulary we see the effects of averaging. Assuming equally probable choices from the 36 words, a vocabulary with an approximate recognition rate of 80% is combined with one whose rate is 26% in the ratios 10/36 and 26/36 respectively. This gives approximately 40% recognition which is roughly equivalent to a branching factor of 2.5. This method of combining branching factors is an approximation, valid only when the recognition rates are near 100% (effective branching factor of 1) and there is no inter-vocabulary ambiguity. There are inter-vocabulary ambiguities; "two" and "u" or "three" and "g", for instance. This would account for the effective branching factor being greater than predicted from the independent results for the two vocabularies.

Consider now the tasks having syntactic restriction. The number of words in their vocabularies range from 17 to 410 while the effective sizes range between 1.46 and 3.54. They seem to be directly related. This relative ordering is expected since large vocabularies have greater potential for ambiguity and therefore would, in

general, have larger effective sizes. An interesting comparison can be made between the chess vocabulary and the Lizard vocabulary. In this case, the 17 words of the Lizard vocabulary have slightly higher confusion than the 25 words of the chess task.

One reason for the large effective vocabulary size of the Lincoln Labs Basic task (9.15) is the fact that it contains words pairs which are almost identical; such as, "to"-"two", "recompute"-"recomputed" and "spectra"-"spectrum". This points to a difficulty in the representation of a vocabulary. Namely, when should two words be considered separate entities. In the "two-to" case, syntax would probablely disambiguate them and the analysis procedures would treat them separately when considering syntax. If "spectra" and "spectrum" appear within the same context and are functionally differentiated, they must remain as two distinct words. On the other hand, if they describe the same semantic notions, then the ambiguity is not one of real concern.

The branching factors for the "grammar only" case fall into the range 7.32 to 20.28. We see that syntactic restriction alone has nearly equalized the difficultly of the Chess, Lincoln Labs Basic and IBM languages. Lizard and VP have larger branching factors, even though they have fewer words in their vocabularies. The Lincoln Labs extended task has the largest branching factor of syntactically constrained languages.

Each of the languages, except IBM's, contain the numbers in one form or another. In the Chess task, the numbers are all single digits indicating rank or file. In Lizard and VP numbers are sequences of digits of indefinite length. They occur in every sentence; in Lizard, this accounts for the branching factor being near ten. In VP numbers occur in approximately 75% of the sentences.

In the Lincoln Lab grammars, numbers come in the general form "one hundred twenty four" but occur rarely.

The largest syntactic branching factor in the table belongs to VPNS. This task has no syntactic constraints, uses the vocabulary of VP and attempts to recognize sentences from VP. This configuration recognizes 90.8% of the words and 62.0% of the sentences.

5.3. Search Space Reduction

One could compute the search space size given this new branching factor in the same manner as was done in chapter 4. A more revealing number is the reduction in search space size. We define the search space reduction ratio for a given branching factor B as the log of (vocabulary size/B)$^{(\text{average sentence length})}$. A table of search space reduction ratios for the tasks investigated is given in Figure 5-4. The column labeled VOC is the search space reduction for B=effective vocabulary size. The column labeled SYN is for B=dynamic branching factor and the third column is for B=effective dynamic branching factor computed in this chapter. This is the total reduction including vocabulary and syntax. The sum of the first two columns is not equal to the third column. This is to be expected since the two interact. In all cases, the vocabulary restriction is greater than the syntactic reduction. The vocabulary provides much more constraint than the syntax for the first three tasks. For the last three tasks, the ones with large vocabularies, the syntax provides much more restriction.

log[SEARCH SPACE REDUCTION RATIOS]

TASK	VOC	SYN	TOTAL
CHESS	33.15	14.29	36.79
LIZ	20.97	5.27	21.54
VP	36.53	14.57	39.88
IBM	54.66	41.20	63.38
LLBAS	49.66	35.26	57.28
LLEXT	61.25	38.75	73.81

VOC - Vocabulary Alone
SYN - Syntax Alone
TOTAL - Total reduction

Figure 5-4. Search Space Reduction Ratios.

6. COMPLEXITY IN CONNECTED SPEECH - A General Model

A best behavior model for the analysis of ambiguity in connected speech was exhibited in chapter 5. In this chapter the limitations of that model are discussed. Then, a general model for complexity in connected speech is developed. This model represents "worst" behavior in the sense that it attempts to predict the ambiguity faced in an errorful recognition.

The major limitation of the restricted model developed in chapter 5 is that it assumes that almost all the recognition proceeds without error. That is, the process moves from one correct state, say p, to another correct state; the ambiguity encountered being a function solely of the words which may follow state p. The consequences of this assumption are that the unit of time depends upon the choice of the representation of the lexicon and syntactic network. For the purposes of the previous chapter, the word was chosen as the fundamental network path length. The choice could just as well have been syllables; the model applies in this case also. Another consequence is that boundaries are assumed to be detected without error. Experience with speech understanding systems indicates that nothing is ever certain. In particular, there is an uncertainty about which state is the "correct" state. This uncertainty means that the number of words which may appear next in the speech is greater than that given by a single sub-vocabulary, as in the previous model.

6.1. Ambiguity in Connected Speech

It will be worthwhile, at this point, to consider various situations in which the correct state becomes uncertain. The obvious way is when the sub-vocabulary of a

state contains two (or more) ambiguous words. For example, "and" and "ant" or "we" and "the". Similarily, a phrasal ambiguity, such as "into" being confused with "in two", may lead to an incorrect state. The obverse of phrasal ambiguity is when a word is ambiguous with an initial substring of another word. This may happen for "in" and "into" or "an" and "and", for example. This case differs from the previous two cases in that the incorrect state may now be within a word. Such errors may have already occurred, in which case, an incorrect state leads to another incorrect state. Suppose that the first three syllables of "accumulate" had been matched with some phonetically similar acoustic sequence. In this case, the last syllable, "-late", may be confused with any syllable of another word, say "late-ness". All of these types of errors may occur when recognizing connected speech; and, as shown in the last example, they may compound. Some examples from the Harpy Recognition System[Lowerre, 1976] will show the kind of errors that may occur.

Correct: Gamma becomes negate epsilon.
Recognized: In mod becomes negate epsilon.

Correct: What is (s)even plus eight.
Recognized: One is one plus beta.

6.2. General Ambiguity Model

The approach, as it differs form the previous chapter, will be to consider the opposite end of the spectrum, i.e. "worst" case behavior. And then to consider modifications to the model which approach more nearly the actual situation. We want to measure the ambiguity given that the correct state is not known with certainty. Let p be the correct state and e be a state which may have been reached because of recognition errors. Now, let x be a path leaving state p and y be a path leaving state e. If x is the path which should have been followed and x and y are phonetically

similar paths, the uncertainty concerning the "correct" state (given that x is the correct path) will be reduced very little. If, on the other hand, all paths leading from possible error states are dissimilar phonetically from x, the uncertainty about the correct path will be greatly reduced. Thus, what we want to measure is the ambiguity of all paths leading from possible error states; keeping in mind that an error state may be in the middle of a word.

Let p be the correct state. The absolute worst case would be where every other state was a possible (error) state and for every path leaving state p there was an identical path leaving every other state. If all paths from all states are identical, the next word of the utterance gives no information about which state is in error. Thus, if all states were equally probable before recognition of the word, they would also be equally probable afer recognition. The uncertainty or ambiguity under these conditions would be the log of the number of states. Clearly, allowing all states as being possible is unrealistic and would involve a great deal of computation. This problem can be rectified by application of the following heuristic knowledge: the difference in the number of syllables in a mistaken recognition and the correct sentence is generally 2 or less; and furthermore, at any point of recognition, the number of syllables, counting from the utterance beginning(or ending) usually differs from the misrecognized number of syllables by at most two. Using syllables as the unit of time, the number of paths which must be matched may be pruned in the following manner. Assume each path in the network represents a syllable. Let t be the number of syllables which have been considered by the recognizer. For each (syllable) time t, the set of allowed, correct states may be determined in a manner analogous to computing state probabilities in chapter 4. Formally, let T(t) be the set containing states for which there exists a path of length t from the initial state. Then,

$T(0) = \{ \text{initial state} \}$

$T(t) = \{ s \mid \exists\, r \text{ such that } r \in PREV(s) \text{ and } r \in T(t-1) \}$

Comparison of this with the computation of p(s|t) in chapter 4 should convince the reader that an equivalent definition for T(t) is

$T(t) = \{ s \mid p(s|t)>0 \}$

Define S(t) to be all states in T(t-1), T(t) and T(t+1). Call the set of paths leading from states in S(t) the super-vocabulary (at time t). What is to be determined is the ambiguity of super-vocabularies averaged over all states and times. This is done by computing the effective vocabulary size for each super-vocabulary and using this as the branching factor in the equations of chapter 4.

6.3. Discussion of Results

The results of the computation of "worst" case branching factor are shown in figure 6-1. For reference, the "best" case branching factor is also shown. This computation was done for the Chess, Lizard and voice programming tasks. The larger tasks require so much computation as to be unfeasible at the present time. As expected, the "worst" case branching factor is the larger of the two for all cases. For the Lizard task the "worst" case branching factor is approximately twice that of the "best" case. This would indicate that a recognizer which has made an error requires twice as much information in order that it return to the correct path. The branching factor for the Voice Programming task has increase from 1.3 to 7.6. Most of the sentences in this language contain number of the form <DIGIT> <DIGIT> <DIGIT> ... <DIGIT>. For example "Store one one one". If, at some point in the recognition, there

TASK	BRANCHING FACTORS Restricted Model	General Model
Lizard	1.46	3.17
Voice Prog.	1.28	7.63
Chess	1.09	3.92

Figure 6-1. Comparison of "best" and "worst" case branching factors.

TASK	WORDS Number	BF	SYLLABLES Number	BF
Lizard	17	1.46	25	1.50
Voice Prog.	37	1.28	52	1.53
Chess	25	1.09	33	1.11

Figure 6-2. Comparison of "best" case analysis for words and syllables.

is some uncertainty at to whether n or n+1 syllables have been seen, the next step of recognition provides very little help in shifting the recognition toward the correct path; in this case, the ambiguity is high. In fact in this particular case, the system would not recover without semantic of other higher level knowledge. Using syllables as the unit of time means that shorter phone strings are matched at each time interval. Shorter strings, in general, imply greater ambiguity. This effect must be considered when comparing the "best" and "worst" case results. In figure 6-2, the effective vocabulary size for the syllable vocabularies has been added. The results show the increase due to using syllables is small relative to the increase due to the "worst" case.

7. RESULTS OF LANGUAGE ANALYSIS

In this chapter we present and discuss the results of language analysis. The results are summarized in Figure 7-1. The first two columns of this table give the task name and the number of words in the vocabulary of the task. The columns to the right contain branching factors under various conditions. The effective branching factor for the vocabulary, without the effects of syntactic restriction, is shown the column labeled "vocabulary only". It is the same as the effective vocabulary size described in chapter 3 and represents the average number of words retrieved in a lexical match per word spoken. Thus, for the 10 digits, 1.19 words would appear, on the average, for each word spoken. The column marked "grammar only" gives the average branching factor considering syntax, but disregarding the effects of lexical ambiguity. This branching factor, described in chapter 5, represents the average fan-out of the syntax; or, the average number of words which may follow another word in an utterance. This column is the same as the vocabulary size for the first four tasks since any word may follow any other word. For the tasks with syntactic constraints, this branching factor ranges from 7.32 words to 20.28 words. The last column contains the effective branching factor for the combined effects of lexical ambiguity and syntactic constraint discussed in chapter 6. A brief description of each of the tasks preceeds a detailed discussion of the results.

7.1. DESCRIPTION OF THE TASKS

Appendix C contains descriptions of the languages analyzed in this thesis. Each description consists of a definition of the syntax of the language and a dictionary for

Task	Number of Words in Vocabulary	BRANCHING FACTORS Vocabulary Only	Grammar Only	Vocabulary and Grammar
PHONES	33	20.10	33	20.10
DIGITS	10	1.19	10	1.19
ALPHABET	26	3.87	26	3.87
ALPHA-DIG	36	3.41	36	3.41
CHESS	25	1.46	7.36	1.09
Lincoln Labs				
Basic	236	2.43	9.15	1.20
Extended	410	3.54	20.28	1.34
IBM	250	2.31	7.32	1.09
LIZARD	17	1.55	9.32	1.46
VP	37	1.70	10.82	1.28
VPNS	37	1.70	37.00	1.70

Figure 7-1. Results of Language Analysis.

it's vocabulary. Dictionaries give the allowed pronunciations for each word in the vocabulary. The first four tasks are not truely languages, but are sets of words we wished to analyze. They have been given a simple syntactic description which allows any word to follow any other word.

PHONS: is a language consisting of a set of 33 phones. Describing the phones as a language makes possible the same analysis as for any other vocabulary. This means we can calculate the effective vocabulary size for the phones.

DIGITS: This vocabulary is the 10 digits. It was included because it was one of the first vocabularies used in speech recognition. It is still used, although usually for comparative purposes.

ALPHABET: This vocabulary is the spoken letters of the alphabet. It is highly ambiguous phonetically and is therefore a good test case.

ALPHA-DIGIT: Is the combination of the 10 digits and the 26 letters. Having this vocabulary allows one to evaluate the effect of combining two vocabularies.

CHESS: The original Hearsay I chess task language. It has a vocabulary of 25 words.

LIZARD: Lizard is a small voice programming language with a vocabulary of 17 words. It has been used with the HARPY speech recognition system.

VP: This language is also a voice programming language. It has been used by both the Hearsay I system and the HARPY system. It is richer in it's syntax than Lizard and contains 37 words. This language has been used extensively as a test case for the HARPY Speech Understanding System in a mode where any word can follow any other word; i.e. there is no syntactic support. The results for this

configuration of the language are shown in the last line of the table under VPNS.

IBM: This is the IBM "New Raleigh" Language. It describes syntactically correct English-like sentences with little or no semantic interpretation.

LLBAS: A language developed by Lincoln Labs for use with their speech recognition system. It's task is displaying and controlling acoustic data. There are 236 words in it's vocabulary.

LLEXT: An "extended" version of LLBAS containing 410 words.

7.2. DISCUSSION OF RESULTS

We will first consider the tasks in order and then general aspects of the complete table. Recall that the phone task vocabulary was just the set of phones. The effective vocabulary size obtained is 20. This means that every phone, on the average, matches uniformly to 20 phonetic labels. It must be remembered that this is for isolated phones without syntactic support, or even a surrounding lexical context. Even so, this value seems rather high. This quantity has been computed from actual counts from the BBN speech recognition system [Makhoul, 1975]. The value for their system, which uses 67 different phoneme types and 83 acoustic classifications, is 4 labels/segment. If this figure were used as a standard, it says that the computation of H(A/B) is roughly two and one-half times larger than it should be. If anything, this implies that our model accounts for more variability in the phones than is really there; that is, it is biased away from high quality, well articulated speech. We intended this to be the nature of the system. Also, bear in mind that the models were designed for relative comparisons.

For the 10 digits, the effective vocabulary size is 1.19. The interpretation here is that six words will be retrieved for every five words spoken and one of them is obviously wrong. This corresponds, roughly, to a recognition rate of 83%. Currently, speech recognition systems have very little trouble recognizing the digits spoken in isolation. Again, we see that if the model is biased, it is biased toward greater variability. We feel that this is actually an advantage of the model; for, given the relative soundness of the model, the differences between vocabularies are enhanced.

The spoken alphabet exhibits an effective vocabulary size of 3.87 words. This is reasonable, particularly when compared to the digits, since the spoken alphabet is highly ambiguous.

In the alphabet-digit vocabulary we see the effects of averaging. Assuming equally probable choices from the 36 words, a vocabulary with an approximate recognition rate of 80% is combined with one whose rate is 26% in the ratios 10/36 and 26/36 respectively. This gives approximately 40% recognition which is roughly equivalent to a branching factor of 2.5. This method of combining branching factors is an approximation, valid only when the recognition rates are nearly 100% (effective branching factor of 1) and there is no inter-vocabulary ambiguity. There are inter-vocabulary ambiguities; "two" and "u" or "three" and "g", for instance. This would account for the effective branching factor being greater than predicted from the independent results for the two vocabularies.

Consider now the tasks having syntactic restriction. The number of words in their vocabularies range from 17 to 140 while the effective sizes range between 1.46 and 3.54. They seem to be directly related. This relative ordering is expected since large vocabularies have greater potential for ambiguity and therefore would, in

general, have larger effective sizes. An interesting comparison can be made between the chess vocabulary and the Lizard vocabulary. In this case, the 17 words of the Lizard vocabulary have slightly higher confusion than the 25 words of the chess task.

One reason for the large effective vocabulary size of the Lincoln Labs Basic task (9.15) is the fact that it contains words pairs which are almost identical; such as, "to"-"two", "recompute"-"recomputed" and "spectra"-"spectrum" This points to a difficulty in the representation of a vocabulary. Namely, when should two words be considered separate entities. In the "two-to" case, syntax would probablely disambiguate them and the analysis procedures would treat them separately when considering syntax. If spectra and spectrum appear within the same context and are functionally differentiated, they must remain as two distinct words. On the other hand, if they describe the same semantic notions, then the ambiguity is not one of real concern.

The branching factors for the "grammar only" case fall into the range 7.32 to 20.28. We see that syntactic restriction alone has nearly equalized the difficultly of the Chess, Lincoln Labs Basic and IBM languages. Lizard and VP have larger branching factors, even though they have fewer words in their vocabularies. The Lincoln Labs extended task has the largest branching factor of syntactically constrained languages.

Each of the languages, except IBM's, contain the numbers in one form or another. In the Chess task, the numbers are all single digits indicating rank or file. In Lizard and VP numbers are sequences of digits of indefinite length. They occur in every sentence; in Lizard, this accounts for the branching factor being near ten. In VP numbers occur in approximately 75% of the sentences.

In the Lincoln Lab grammars, numbers come in the general form "one hundred twenty four" but occur rarely.

The largest syntactic branching factor in the table belongs to VPNS. This task has no syntactic constraints, uses the vocabulary of VP and attempts to recognize sentences from VP. This configuration recognizes 90.8% of the words and 62.0% of the sentences[Lowerre, 1976].

8. CONCLUSION

This dissertation describes a general model for the analysis of languages for man-machine communication. It the first known study of ambiguity at all levels of recognition and represents the best analytical tool ~~we have~~, to date, for the design of languages. This chapter presents a summary of the results and indicates directions for future research.

8.1. Contributions

8.1.1 The Overall Model

The model unifies the concepts of ambiguity and restriction. ~~This~~ is done by expressing each as a branching factor, a notion which is easily understood and visualized. Ambiguity increases the branching factor while restriction reduces it. Using branching factor has the advantage that an effective search space size may be computed for any language. Further, since ambiguity and syntactic restriction are expressed in a uniform way, the effect of one with respect to the other may be evaluated by considering search space reduction ratios.

The model is useful for comparing the relative complexities faced by speech understanding systems. Effective vocabulary size provides a way of measuring the complexity in isolated word recognition while effective search space size measures language complexity. Thus, the performance of ~~two~~ systems may be contrasted by using these measures. Previously, this could be done only if the ~~two~~ systems had been tested using the same data; a situation which occured rarely.

END

Analyzing and anticipating the ambiguities encountered in a specific language is useful for language design and benchmarking. Language design is discussed in the section on future work. Benchmarking means deciding whether the expected performance of a given task is being achieved. If it is not, examination of the errors which occurred and were not predicted by the model may point out flaws in the system which had gone unnoticed; and vice versa.

8.1.2 Phonetic Ambiguity

The model uses phone-to-phone distance measures as a basis for subsequent analysis. We have indicated several ways these measures might be obtained. The choice of which one to use will depend on what is to be modeled and what type of data is available to the user. Actual counts may be used, provided they are trustworthy. Data may be obtained from either human perceptual or machine recognition studies may be used. We have shown how metrics on the acoustic space can be used. One of these, the Itakura metric, has been used as a basis for the analysis presented in this thesis. Another method of obtaining these measures is by using a theoretical model. We have presented one theoretical model, an articulatory model. The performance of this model is not as good as we had hoped. It appears that the phones must be described in finer detail in order to accurately capture their relative differences. We intend to improve our model and also will look for other work along these lines.

8.1.3 Lexical Ambiguity

In computing lexical ambiguity we developed a phone sequence matching algorithm which is easily extendible to phrases. Effective vocabulary size was shown

to be a valid measure of the inherent complexity of a vocabulary. Information theoretic concepts proved useful in this analysis. We feel they are applicable in many other areas of speech understanding systems.

8.1.4 *Syntactic Restriction*

We have exhibited a useful way of viewing syntactic restriction, i.e. dynamic branching factor. This measure of complexity is compatible with the measure of vocabulary complexity. The notion of branching factor has been used in other areas of computer science. When applied in a straight forward way to measure syntactic ambiguity, it is very revealing. We have seen task descriptions which list the number of non-terminals and rules of the grammar; they should also list the average branching factor.

8.1.5 *Language Analysis*

Two models for ambiguity in connected speech were presented: a "best" behavior model and a "worst" behavior model. Both models combine the effects of lexical ambiguity and syntactic restriction. The "best" behavior model measures the ambiguity encountered when most of the recognition proceeds without error. The "worst" behavior model measures the ambiguity faced by an error-prone system. In effect, it indicates the difficulty of returning to the correct path given that the recognition has taken a wrong path.

8.1.6 *The Tasks*

The Chess task[Reddy, et al.,1972; Baker, 1975; Lowerre, 1976] has an effective search space size of 23.31. Its equivalent vocabulary size of 1.46 is the lowest of all

the tasks studied. The effective branching factor for this task is 1.09; also the lowest and the same as for the IBM task.

The Lincoln Labs "extended" task [Forgie, et al., 1974] has the largest search space size, 38.79 It is the most difficult task by all measures except effective branching factor; Lizard and VFNS having larger effective branching factors. The "basic" task, even though its vocabulary contains 236 words, is of roughly the same difficulty as the voice programming task when considering syntactic and effective branching factors.

The IBM "New Raleigh" task [Tappert, 1975; Baker and Bahl, 1975] has an effective search space size of 23.23. Its effective branching factor is 1.09, the same as for the chess task. The syntactic branching factor for this task is 7.32, lowest of all the tasks.

For the Lizard task[Lowerre, 1976], the search space size, 19.56, is the smallest of all the tasks. Its effective branching factor of 1.46, however, is the largest of the languages having syntactic constraints.

The voice programming task, VP [Erman, 1974; Baker, 1975; Lowerre, 1976], has an effective search space size of 28.24. This task has the largest syntactic branching factor of the medium sized languages. VP with no syntax has the highest syntactic branching factor.

The important contribution of this thesis is that it provides a way to characterize the relative difficulties and accomplishments of different speech understanding systems. Vocabulary size is not a good measure of lexical complexity; some other measure of vocabulary size, normalized for relative ambiguity would be

better. The number of production rules is not a useful measure of grammatical complexity. In fact, quite the opposite may be true; more rules imply more constraint. Some other measure, such as the average number of alternatives at each choice point would be better. Investigators in the area of speech understanding should reference their results to some standard. This thesis presents some useful measures.

8.2. Directions for Future Research

With the generation of any large system, particularly in a new area, many new ideas for improvements are spawned and many inviting avenues are left unexplored. This investigation was no exception. Possible improvements to the model are outlined below.

1. Improvement of the theoretical phonetic ambiguity model will be necessary in order for it to be used as a basis for the lexical and phrasal model. Until such time, the acoustic similarity metrics described should be adequate.

2. Although the model provides a particular solution to the juncture ambiguity problem, more detailed used of phonological rules should lead to a more precise model.

3. Analysis of the ambiguities encountered in segmentation and their implications for phonetic ambiguity should lead to a better model.

4. The model assumes that Context-Free languages, as used in speech understanding systems, can be represented, for all practical purposes, by a finite state approximation. In doing this, some small amount of restrictive power may be lost. While this is not considered a serious problem, further investigation into the nature of its effects should be considered.

5. Semantic ambiguity happens when two sentences are phonetically similar enough that one may be recognized as the other (or they may be both recognized, with a match score for each, by some systems) and the two sentences cannot be disambiguated by semantics. Conversely, semantics may apply constraints to the vocabulary and syntax which would eliminate ambiguous sentences from being considered. The notion of branching factor accommodates either viewpoint. Analysis should be done at this level also, although we have no specific ideas about how it could be done. It should be investigated to whatever extent possible.

8.3. Implications for Language Design

Given that a reasonable analytical tool is available, a fruitful area for future research is the design of languages for man-machine communication. Designing languages would include, but not necessarily be limited to, the following possibilities:

1. Reducing the ambiguity of a language by altering the vocabulary and syntax of the language or by redefining the task. Sometimes alteration of the vocabulary and syntax may be hindered by standard or accepted usage. This would be true of the numbers and the chess task. At other times, there are free choices; as with the names "ALPHA", "BETA", "GAMMA", "DELTA", "EPSILON" in the voice programming language.

2. Tailoring a task and language to some predefined constraints. For instance, it would be desirable to know just how much ambiguity could be tolerated by a system whose processor was a mini-computer with restricted memory and fixed instruction time. This aspect will become increasingly important

as the use of speech understanding systems grows and new tasks are undertaken.

In order to do design of languages, one must understand the ambiguities involved. The results of the analysis presented in this dissertation provide this information.

REFERENCES

Bahl, L.R., J.K. Baker, P.S. Cohen, N.R. Dixon, F. Jelinek, R.L. Mercer, and H.F. Silverman (1976), "Preliminary Results on the Performance of a System for the Automatic Recognition of Continuous Speech", **Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing**, Philadelphia, April 1976, 425-429.

Baker, J.K. (1975), Stochastic Modeling as a Means of Automatic Speech Recognition, Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania, April 1975.

Baker, J.K. and L.R. Bahl (1975), "Some Experiments in Automatic Recognition of Continuous Speech", **Proceedings IEEE Computer Conference 75**, September 1975, 326-329.

Erman, L.D. (1974), An Environment and System for Machine Understanding of Connected Speech, Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania, May 1974.

Feldman, J. and D. Gries (1968), "Translator Writing Systems", **Communications of the ACM 11, 2** (February), 1968, 77-113.

Forgie, J.W., and C.D. Forgie (1959), "Results Obtained From a Vowel Recognition Computer Program", **Journal of the Acoustical Society of America, 31**, 1480-1489, November 1959.

Fo[illegible]e, J.W. et al. (1974), Speech Understanding Systems: Semi-Annual Report, Lincoln Labs, MIT, Lexington, Mass., May 1974.

Fu, K.S. and T. Li (1969), "On Stochastic Automata and Languages", **Information Sciences**, Vol. 1, 403-420, 1969.

Goldman, Stanford (1953), **Information Theory**, Prentice-Hall, New York, 1953.

Itakura, F. (1975), "Minimum Prediction Residual Principle Applied to Speech Recognition", **IEEE Transactions on Acoustics, Speech, and Signal Processing, 23**, February 1975, 67-71.

Ladefoged, P. and D.E. Broadbent (1970), "Information Conveyed by Vowels", **Journal of the Acoustical Society of America, 29**, 98-104, January 1957.

Lowerre, B. (1976), The HARPY Speech Recognition System, Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, April 1976.

Makhoul, J. (1975), personal communication.

Miller, G.A. and P.E. Nicely (1955), "An Analysis of Perceptual Confusions Among Some English Consonants", **Journal of the Acoustical Society of America, 27**, 338-352, March 1955.

Newell, A., J. Barnett, J. Forgie, C. Green, D. Klatt, J.C.R. Licklider, J. Munson, R. Reddy, and W. Woods (1971), **Speech Understanding Systems: Final Report of a Study Group**, Pub. by North Holland (1973).

Newell, A. (1975), "A Tutorial on Speech Understanding Systems", **Speech Recognition**, Reddy, D.R.(Ed.), Academic Press, New York, 1975.

Peterson, G.E. and H.L. Barney (1952), "Control Methods Used in a Study of the Vowels", **Journal of the Acoustical Society of America, 24**, 175-184, March 1952.

Reddy, D.R., L.D. Erman, R.D. Fennell, and R.B. Neely (1973), "The HEARSAY Speech Understanding System: An Example of the Recognition Process", **Proceedings of the 3rd International Joint Conference on Artificial Intelligence**, Stanford, California, 185-193.

Tappert, C.C. (1975), "Experiments with a Tree-Searching Method for Converting Noisy Phonetic Representation into Standard Orthography", **IEEE Transactions on Acoustics, Speech, and Signal Processing, 23**, February 1975.

Unger, S. (1968), "A Global Parser for Context-free Phrase Structured Grammars", **Communications of the ACM 11, 4** (April), 240-247.

Woods, W.A. (1970), "Transitions Network Grammars for Natural Language Analysis", **Communications of the ACM 13, 10** (October), 1970, 591-602.

Appendix A: Phonetic Ambiguity - Itakura Metric

A.1. Itakura Metric Calculation

The Itakura metric [Itakura, 1975] matches an input signal with stored reference patterns using the distance function:

d(X/a) = c + log[(br) / (a'r)]

where

(xy) means the inner product of two vectors,

X is a segment of the time signal x(1), x(2), ...x(N),

a = 1, a(1), a(2), ...a(p) are the LPC model parameters for the reference pattern,

c = log[(aa)],

b = 1, b(1), b(2), ... b(p) are the modified LPC coefficients computed from a,

r = r(0), r(1), ...r(p) are the autocorrelation coefficients for X,

a' is the vector representing the LPC model for X.

Reference patterns for the phones used in this thesis are given in section A.2. Each pattern has the form:

phone	c				
	1.0	b(1)	b(2)	b(3)	b(4)
	b(5)	b(6)	b(7)	b(8)	b(9)
	b(10)	b(11)	b(12)	b(13)	b(14) ;

These coefficients were derived from the autocorrelation coefficients given in A.3. They have the form:

phone	1				
	1.0	a(1)	a(2)	a(3)	a(4)
	a(5)	a(6)	a(7)	a(8)	a(9)
	a(10)	a(11)	a(12)	a(13)	a(14) ;

The distance, d(X/a), is the logarithm of the conditional probability, p(X/a), that the input signal X was generated by the LPC model defined by a. By comparing the autocorrelation coefficients with the reference patterns, the conditional probabilities p(p1/p2) may be computed for each pair of phones p1 and p2. A matrix of these probabilities is shown in section A.4. This tables contains probabilities which are normalized so that p(x/x)=1.

A.2. Reference Patterns for Itakura Metric

-	.4653838					
	.1000000e1	.9481453	.4112804	.4116420	.1575353	
	.1072501	-.1488018	-.5794935	-.3825964	-.3537391	
	-.4941177	-.3728067	-.4078989	-.2399488	-.4908963e-1	;
V	.1572590e1					
	.1000000e1	-.1468318e1	.8000144	-.6130934	.4426651	
	.1855549	-.6291837	.5498009	-.3225522	.2688221	
	-.2070323	.8631829e-1	.2886842e-1	-.2961992e-1	.2835220e-1	;
N	.1772191e1					
	.1000000e1	-.1646469e1	.1352822e1	-.1012057e1	.5329455	
	-.4720814	.4815684	-.5448401	.7090770	-.6561077	
	.6032893	-.4587166	.2588628	-.1094428	.4696169e-1	;
D	.8561885					
	.1000000e1	-.8415627	.3819256	-.2366808	-.6365611	
	-.6732695e-1	.4177934	-.1559834	.4553004	-.7375370e-1	
	-.2167171	.4425386e-1	-.1321079	.5216035e-1	.8635711e-1	;
AO	.1072732e1					
	.1000000e1	-.5092934	.5514796	.2570911	-.5485894	
	.1184381e1	-.1560861	.6362915	.1875962	-.3419255	
	.3963176	-.9324697e-1	.2159744	.1317259	-.1127955e-1	;
K	.1326324e1					
	.1000000e1	.1168543e1	.1063832e1	.3204955	.1969025	
	.4031753	.2706019	.4424938	-.1306385e-1	-.5727035e-1	
	-.2750050	-.1591132	-.1037082	-.2334523e-1	-.1422657e-2	;
G	.6178884					
	.1000000e1	.6223440	.9874111e-2	-.7767388	-.1045987e1	
	-.5122625	.4416202e-2	.2150669	.3645362	.2138235	
	.2000004	-.1043659	-.5509937e-1	-.7320712e-1	-.5784028e-1	;
S	.2376628e1					
	.1000000e1	.1731307e1	.1218139e1	.6293892	.2289342	
	-.3514551e-1	-.1976203	-.2632045	-.2706624	-.2245494	
	-.2147891	-.1886363	-.1595308	-.8102228e-1	-.2997903e-1	;
T	.1505892e1					
	.1000000e1	.1458097e1	.1369474e1	.7766711	.7120834	
	.3607056	.1188773	-.8224959e-1	-.2224482	-.1560316	
	-.1774207	-.2542178e-1	-.2121245e-1	.3795647e-1	-.1478976e-2	;
AX	.2204932e1					
	.1000000e1	-.1795964e1	.1445459e1	-.1016270e1	.6009594	
	-.2387336	.3375595e-1	.2915856e-1	.3220392e-1	-.1017121	
	.1548753	-.1409085	.1462294	-.9746549e-1	.3219684e-1	;
AE	.2476707e1					
	.1000000e1	-.1768121e1	.1713801e1	-.1533954e1	.1299149e1	
	-.1025217e1	.8195718	-.5684309	.4557624	-.2262559	
	.1996797	-.1198496	.4517272e-1	-.2205637e-1	.3388926e-1	;
IH	.6985018					
	.1000000e1	.2738923	.1002042e1	.4243288	.8639940	
	.6861938	.5040877	.5591835	.2406445	.2765797	
	.3554765	.3171800	.1915672e-1	.6763018e-1	.1357980e-1	;
AA	.7072659					
	.1000000e1	.2753748	.7224589	.3563320	.2824728	
	.8663480	.6489110	.5757259	.2054031	-.4682825e-1	
	.2939971	.1419088	.1955692	.2081737	-.1274257	;

M	.1041473e1					
	.1000000e1	-.1040802e1	.9640988	-.1179369e1	.4358865	
	-.5844040	.8064014	-.4076996	.5097560	-.7423652	
	.2589428	-.3263575	.1954818	.1221869	.4674456e-1	;
EH	.2560478e1					
	.1000000e1	-.1775052e1	.1597873e1	-.1251882e1	.8550688	
	-.5145747	.2762279	-.5831185e-1	.4206885e-1	.4431842e-1	
	-.6470111e-2	-.1147008e-1	.5200001e-2	-.4698264e-2	.2071476e-1	;
W	.1014307e1					
	.1000000e1	-.6719078	.7196769e-1	-.7023114	-.1419251	
	.4006261	-.8557946e-1	.5577567	-.3378162	-.1391835	
	-.8690805e-1	.1774109	.7728362e-1	-.1954272	.9564959e-1	;
NX	.1356551e1					
	.1000000e1	-.1486121e1	.8032778	-.1848480	-.3841723	
	.3758652	-.1768082e-1	-.2055322	.4001247	-.3883483	
	.1378254	.1957673e-2	-.5817021e-1	.1212812	-.4414433e-1	;
L	.5973559					
	.1000000e1	.2350808e-1	.1959081	-.3474107	-.2667801e-1	
	.4557708	.2300012	.9098240	.1685848e-1	.2178133	
	-.1998036	-.1159052	.2923621	.1849393	.2048662	;
UW	.1391596e1					
	.1000000e1	-.1202846e1	.8484092	-.1021389e1	.1987648	
	.1752733	-.1504844	.6591436	-.5427459	.3002487	
	-.5952312	.4059821	-.7546913e-1	.1834871	-.1151178	;
Y	.1646643e1					
	.1000000e1	-.2152640	.6484815	-.1191197e1	-.4156332	
	-.3173563	.2340207e-1	.5350594	-.1295259	.2749016	
	-.2908673	.1482031	-.7192482e-1	.8356436e-1	-.1322980e-1	;
ER	.2868491e1					
	.1000000e1	-.1713589e1	.1122430e1	-.5752222	.7672869e-1	
	.4391078	-.7811429	.7914453	-.6108526	.4479250	
	-.3225888	.1890355	-.6967031e-1	.1219156e-1	.1826608e-3	;
B	.4820763					
	.1000000e1	.7175219	.1511603	-.4202538	-.6599186	
	-.4063518	-.3303471	-.3435398	-.1423548	-.2939424	
	.1343224	.2331645	.2281689	.2199586	-.4870881e-1	;
DH	.5024395					
	.1000000e1	-.6374152	.9040350	-.5291229	.2992921	
	.1652868	.1482031e-2	.2494591	-.6547075e-1	.4329374	
	-.1077068	.1002039	-.2301427	-.5612478e-2	.4580149e-2	;
IY	.2204433e1					
	.1000000e1	-.1102626e1	.1345012e1	-.1294917e1	.4543056	
	-.7928261	.2277431	-.5233127e-1	.1429815	.1357118	
	-.1170542e-1	.5681549e-1	-.4029132e-1	.1701903e-1	-.7327487e-2	;
F	.6091735					
	.1000000e1	.4107559	.8780966	-.1209926	-.1560682	
	.1864959	-.1758721	.4450189	- 2035727e-2	.2408079	
	.1095976	-.5763851e-1	.1499390	-.8230528e-1	.1108176	;
HH	.1183437e1					
	.1000000e1	.3141814e-1	.1017938e1	-.8251846	.2201196e-1	
	-.7516666	.2733571	-.3162514	.4856131	-.3479637	
	.3361489	-.2939970	.2172461	-.8160477e-1	.8194286e-1	;
P	.1339082e1					
	.1000000e1	.6403634	.1526930e1	.6169871	.1186754e1	
	.8069608	.7939511	.6470240	.3327926	.4705566	
	.2325033	.2406440	.1086712	.6998843e-1	.4280503e-1	;

```
OW      .6502480
        .1000000e1   -.1455020    -.5052606    -.2183936     .6560606
        .9477162e-1  -.1433321     .5304350     .1393801    -.3351505e-1
       -.2956620e-1   .6160112e-1  .3498848e-1  .8494031e-1  .1598899    ;
SH      .2699239e1
        .1000000e1    .1691833e1   .1246778e1   .5672762     .9117345e-1
       -.1456324     -.1861294    -.2305883e-1  .6787734e-1  .1791906
        .1534690      .1247145     .6044490e-1  .2628993e-1  .5067975e-2 ;
UH      .2119180e1
        .1000000e1   -.7463413     .2884738    -.1282107e1   .4040748
        .4388554      .1837626     .1617227e-1 -.4935050     .1932944
       -.4208080e-1   .1475354    -.7527767e-1  .7503602e-3  .6693101e-2 ;
AH      .1323664e1
        .1000000e1   -.7598235     .4903862    -.1339286e1   .3519908
        .2142586      .4420243     .1161817    -.5935806     .1170325e-1
       -.1197412      .4143636    -.3749145e-1  .6562085e-1 -.1098726    ;
```

A.3. Autocorrelation Vectors for Phone Reference Patterns

-	1				
	.1000000e1	-.2288125	.3144131	.5666456e-1	.1614015
	.1864762	.4349749e-1	.3764852	.781[illegible]5e-1	.1371318
	.2797422	.5251361e-1	.3055862	.633[illegible]1e-1	.2218621
V	1				
	.1000000e1	.5452471	.1303280	-.50[illegible]633e-1	-.5032854
	-.3609834	.5226166e-1	.2269147e-1	.2075819	.2787059
	-.9801593e-1	-.1737188	.2323351	-.4489989	-.2368247
N	1				
	.1000000e1	.1884187	-.3950262	.3845842	.5868284
	-.1538444	-.2341701	.2543882	.1714968	-.2714685
	-.2289182	.1599807	.2630599e-1	-.3364862	-.1518713
D	1				
	.1000600e1	.4697946	-.1329718e-1	.3332305	.6452802
	.2134794	-.1889928	.3183799e-1	.2084580	-.7240400e-1
	-.2839559	-.1184846	.988[illegible]e-2	-.2329844	-.3224474
AO	1				
	.1000000e1	.6136462	.2477254	.6492505e-1	-.2798560
	-.5964925	-.5846240	-.5572341	-.2647501	.5985912e-1
	.7445676e-1	.1550780	.3015594	.1579097	.6943268e-1
K	1				
	.1000000e1	-.2443574	-.7759651	.5322135	.4177193
	-.6273410	-.4067868e-2	.4485359	-.2256464	-.2399533
	.3516739	-.8575877e-2	-.2972350	.1481500	.2110758
G	1				
	.1000000e1	.3196531	.1996177	.4710796	.6013976
	.2520509	.1196105	.3514191	.2943729	.1060434
	.2695171e-1	.2377262	.1264296	.5253414e-2	.5614547e-1
S	1				
	.1000000e1	-.4747483	-.4846925	.8304715	-.3148533
	-.4076372	.6513032	-.3190336	.2769919	.5483202
	-.3756318	-.1042743	.6106206	-.4260494	-.2048856
T	1				
	.1000000e1	-.4020670	-.5610065	.7712137	-.1473374
	-.5236049	.4667192	.8076257e-1	-.3723647	.1942898
	.1413873	-.2217581	.5203089e-1	.5920076e-1	-.6631611e-1
AX	1				
	.1000000e1	.6998251	.3162055	.1764444	.1375339e-1
	-.8522591e-1	-.1308032	-.2881705	-.3854576	-.4417292
	-.5816178	-.5916872	-.3823164	-.1071516	.9075869e-1
AE	1				
	.1000000e1	.6073816e-2	-.2984334	.4197423	-.4369415e-1
	-.1438057	.5539806e-2	-.2532939	-.3203958e-1	-.3049655
	-.4768901	.2918202	.1596943	-.3403801	.7956751e-1
IH	1				
	.1000000e1	.2842963	-.1106172	-.4212182e-1	-.4999777
	-.3596229	.1335819	-.1779676e-1	.9935820e-1	.1692002
	-.3698094	-.3045219	.1064142	.5760817e-1	.3318786
AA	1				
	.1000000e1	.6409851	.2493692	.5459470e-1	-.1844885
	-.5492872	-.7374097	-.6377681	-.3757167	-.1648717
	.2025018e-1	.2811722	.4589548	.5124494	.5130397

```
M       1
          .1000000e1    .3347370      -.6528271e-1   .4173726      .5024565
          .1076337     -.3709719e-1    .2084938      .3490489      .1339645
         -.6879224e-1   .2174488       .1749609     -.2697051     -.1350399    ;
EN      1
          .1000000e1   -.1502035      -.5871161      .5424689      .1165730
         -.4295389      .1772078       .1308788e-1  -.2317347      .4275320e-1
         -.1903142      .1838545e-2    .3649301     -.2809462     -.2606403    ;
W       1
          .1000000e1    .8940516       .7268838      .5269770      .2459462
         -.1902803e-1  -.2062547      -.3533340     -.3986995     -.3465359
         -.2707974     -.1690156      -.3543126e-1   .4600482e-1   .5961581e-1  ;
NX      1
          .1000000e1    .4854344       .2157816e-1   .3001680      .3446198
         -.7762506e-1  -.1405668       .2569860e-1  -.8341972e-1  -.1403197
         -.1999288e-1  -.1024013      -.2986752     -.2490006     -.1076794    ;
L       1
          .1000000e1    .5838114       .4136782      .2161084     -.2881899
         -.3858951     -.5060237      -.6296178     -.2781713     -.2457883
         -.6084481e-1   .1603594      -.5265562e-2   .5658329e-1   .3754233e-1  ;
UW      1
          .1000000e1    .5051369       .2158546      .4369369      .8202079e-1
         -.2020248      .6066125e-2   -.1447745     -.2657272      .7029032e-1
          .4551173e-1  -.3149253      -.2616310     -.1763368     -.3781573    ;
Y       1
          .1000000e1   -.7134527e-1   -.6863466      .4319392      .5646289
         -.3014307     -.2453590       .3201644      .1684785     -.2498311
         -.7298993e-2   .1809839      -.1902343     -.1239536      .2263208    ;
ER      1
          .1000000e1    .7941442       .3456068     -.5847240e-1  -.2364898
         -.1556628      .1837145e-1    .6395827e-1  -.6164988e-1  -.2753992
         -.4659607     -.5143643      -.3968895     -.2005946     -.5525432e-2  ;
B       1
          .1000000e1    .2753762       .4017322      .4926546      .4805494
          .3947312      .3385788       .4048386      .2162104      .3918326
          .1656677      .1814504       .1653026      .6175504e-1   .1394349    ;
DH      1
          .1000000e1    .1823235e-1   -.4504184      .2955278      .1696501
         -.3662128     -.1188584       .1481993     -.9306267e-1  -.2467663
         -.1879319e-1   .1442244       .1103181     -.1153363e-1   .9409250e-2  ;
IY      1
          .1000000e1   -.7784000e-1   -.6772431      .1281748      .2821922
          .2959362     -.7145900e-1   -.6005622      .2004710      .5430301
         -.2222094     -.2508274      -.5902641e-1   .3964077e-1   .3197732    ;
F       1
          .1000000e1   -.2338083      -.6168330      .4312937      .2064127
         -.3201671      .1865733e-1    .1125261     -.1204839e-1  -.9278687e-1
          .3837766e-1   .4497844e-1   -.9432520e-1   .9051779e-1   .1227169e-1  ;
HH      1
          .1060900e1   -.1556051      -.7924001      .3265335      .4916686
         -.2876062     -.2355194       .1837326      .7836092e-1  -.6801572e-1
         -.1929621e-1   .3719271e-1   -.6559417e-1   .2611265e-1   .1066373    ;
P       1
          .1000000e1   -.4887110e-2   -.5266492      .2006876      .2037890
         -.2808329     -.5012242       .1043945      .4939695     -.2938778
         -.2790242      .4500421       .2149402     -.2538286     -.1501057    ;
```

```
OW      1
          .1000000e1    .5995029     .3328667     .9707050e-1  -.5099286
         -.6220916     -.5173987    -.5552074    -.1119659      .2455174
          .2667000      .4487311     .3075776    -.8687014e-1  -.1961242    ;
SH      1
          .1000000e1   -.1455001    -.8664353     .2948325      .6482193
         -.2613658     -.5284975     .1703847     .5616904     -.2060202
         -.5770152      .3254668     .4899809    -.4148259     -.3253974    ;
UH      1
          .1000000e1    .6640012     .4080572     .2550887     -.3291388
         -.5763466     -.5733147    -.6978378    -.4178742     -.8756537e-1
          .6886448e-1   .4073686     .4680411     .2920590      .2459644    ;
AH      1
          .1000000e1    .2282843     .8734779e-1  .5485922     -.5553988e-1
         -.4469023e-1   .7842994e-1 -.3295973     .4763085e-1  -.2581209e-1
         -.5576211     -.1670417    -.1703546    -.5275783     -.1783041    ;
```

A.4. Phonetic Ambiguity Matrix - Itakura Metric

	-	P	B	T	D	K	G	F	V	DH	S	SH	HH	M	N	NX
-	100	7	59	11	23	13	45	22	7	23	7	4	15	15	6	10
P	23	100	16	16	16	21	17	31	7	49	5	3	16	29	16	12
B	30	3	100	4	35	6	77	16	11	22	2	1	21	37	15	19
T	17	28	23	100	6	47	15	42	8	32	41	26	34	16	10	6
D	10	3	29	3	100	4	52	15	12	25	1	1	16	59	57	65
K	17	24	21	44	14	100	21	59	4	32	33	39	29	13	10	15
G	26	4	70	4	50	5	100	19	13	28	2	2	23	42	20	27
F	40	50	48	49	23	64	48	100	13	65	25	21	57	33	20	18
V	8	7	8	3	8	3	5	8	100	16	1	1	4	15	17	18
DH	39	48	36	19	43	28	43	59	20	100	7	5	34	46	39	39
S	5	9	10	40	2	25	6	14	4	10	100	48	10	6	3	2
SH	4	17	6	9	4	37	10	21	1	12	21	100	24	11	7	4
HH	19	30	29	30	21	60	33	61	5	35	21	25	100	35	34	19
M	18	4	38	4	43	6	38	17	13	28	2	1	20	100	56	40
N	12	5	27	4	48	6	42	21	8	34	2	1	21	58	100	61
NX	10	5	21	3	57	5	24	16	20	30	1	1	13	50	67	100
W	2	1	3	0	29	1	8	3	27	7	0	0	2	10	19	28
L	6	11	6	2	14	4	8	9	26	24	1	1	4	16	10	11
Y	9	5	18	5	13	7	24	20	3	19	2	2	30	23	22	12
UW	6	4	12	2	25	3	12	10	34	20	1	1	7	35	35	39
UH	1	2	1	0	3	1	1	1	11	6	0	0	1	4	5	4
OW	6	11	4	2	9	3	4	6	32	17	1	0	3	20	11	12
AO	5	19	4	2	15	3	5	7	17	21	1	0	3	15	12	25
AA	7	41	5	2	27	5	9	9	18	38	1	1	4	18	13	32
AH	7	5	17	3	10	4	18	14	22	23	1	1	10	33	21	10
ER	1	1	2	0	3	1	2	2	39	4	0	0	1	3	6	10
AE	8	16	11	4	8	6	17	20	10	33	1	1	12	17	15	7
EH	9	17	12	7	8	11	14	24	6	34	3	2	18	20	41	12
IH	19	53	15	10	12	9	10	18	36	38	3	2	9	18	20	20
IY	5	6	13	6	5	11	10	17	3	8	3	4	34	20	11	4
AX	4	5	7	1	16	2	10	8	37	19	1	1	5	12	28	33

	W	L	Y	UW	UH	OW	AO	AA	AH	ER	AE	EH	IH	IY	AX
-	16	20	9	9	5	25	11	16	12	2	3	2	15	4	3
P	12	24	12	15	14	9	18	38	37	4	31	22	32	11	14
B	25	16	16	14	8	21	10	9	19	3	5	4	8	8	7
T	10	14	25	21	15	9	11	13	23	3	14	10	19	17	8
D	25	18	32	25	8	12	14	10	29	8	18	19	8	26	26
K	7	12	35	14	6	4	16	16	15	3	5	6	11	10	7
G	29	19	25	19	12	18	13	11	26	4	7	6	9	12	10
F	21	36	40	28	13	15	28	39	31	6	14	14	38	22	16
V	14	13	3	18	8	25	16	15	17	35	14	16	17	3	51
DH	26	44	25	34	18	22	41	51	36	10	31	28	51	22	28
S	4	3	11	6	6	5	6	6	10	1	2	2	12	4	2
SH	2	3	24	2	2	1	8	9	10	1	5	6	5	11	3
HH	9	16	38	14	5	6	23	28	18	3	17	16	19	28	14
M	24	18	25	28	13	13	14	11	38	7	20	18	9	19	23
N	13	18	33	20	11	8	18	14	28	6	32	39	10	34	27
NX	26	26	15	44	11	16	21	15	29	14	26	31	12	14	47
W	100	8	4	37	33	12	4	3	19	24	8	9	2	3	22
L	35	100	5	35	19	67	20	26	25	8	8	9	23	3	12
Y	5	9	100	11	5	3	10	10	20	2	16	22	7	51	11
UW	42	23	9	100	15	22	15	10	35	21	19	24	9	7	43
UH	12	13	1	14	100	14	7	5	6	6	7	[illegible]	4	1	11
OW	28	61	3	36	24	100	26	26	20	9	5	10	20	2	15
AO	29	36	3	27	11	26	100	54	19	9	7	8	34	2	13
AA	33	40	5	28	12	25	67	100	22	9	12	13	57	4	19
AH	24	27	14	37	29	24	15	14	100	8	31	26	13	13	28
ER	4	3	1	4	2	5	3	3	3	100	5	5	3	1	33
AE	9	19	11	12	7	10	12	17	13	5	100	47	21	16	18
EH	7	11	15	12	6	6	20	13	11	4	60	100	16	21	17
IH	19	18	5	19	10	27	27	39	23	12	16	15	100	6	26
IY	6	7	36	12	1	3	9	12	6	3	8	15	9	100	15
AX	17	15	5	18	6	19	16	12	15	34	24	26	13	6	100

Appendix B: Phonetic Ambiguity - Articulatory Model

B.1. Articulatory Features and Allowed Values

1. Vocal Tract Closure	O- open C- closed or constricted T- turbulent
2. Vocal Chords	V- vibrating (voiced) U- not vibrating (unvoiced)
3. Nasal Cavity	O- open C- closed
4. Tongue Position	B- back C- central F- front
5. Tongue Height	L- low M- medial H- high
6. Tongue Tip	M- moving N- not moving
7. Lips	N- normal C- closed R- rounded

B.2. Definition of the Phones in terms of their Feature Values

IY	VOICED	OPEN		FRONT	HIGH		
IH	VOICED	OPEN		FRONT	HIGH		
EY	VOICED	OPEN		FRONT	MID		
EH	VOICED	OPEN		FRONT	MID		
AE	VOICED	OPEN		FRONT	LOW		
AA	VOICED	OPEN		BACK	LOW		
AH	VOICED	OPEN		CENTRAL	MID		
AO	VOICED	OPEN		BACK	LOW		
OW	VOICED	OPEN		BACK	MID		
UH	VOICED	OPEN		BACK	HIGH		
UW	VOICED	OPEN		BACK	HIGH		
AX	VOICED	OPEN		CENTRAL	MID		
IX	VOICED	OPEN		FRONT	HIGH		
ER	VOICED	OPEN		CENTRAL	MID	TIP MOVEMENT	
AW	VOICED	OPEN		BACK	LOW		
AY	VOICED	OPEN		BACK	LOW		
OY	VOICED	OPEN		BACK	HIGH		ROUNDED
Y	VOICED	OPEN		CENTRAL	HIGH	TIP MOVEMENT	ROUNDED
W	VOICED	OPEN		CENTRAL	MID		ROUNDED
R	VOICED	OPEN		CENTRAL	MID		CLOSED
L	VOICED	OPEN		CENTRAL	MID	TIP MOVEMENT	
M	NASALIZED	CLOSED	NASAL	FRONT	MID		CLOSED
N	NASALIZED	CLOSED	NASAL	CENTRAL	HIGH	TIP MOVEMENT	
NX	NASALIZED	CLOSED	NASAL	BACK	LOW		
P	UNVOICED	TURBULENT		FRONT	MID		CLOSED
T	UNVOICED	TURBULENT		CENTRAL	HIGH	TIP MOVEMENT	
K	UNVOICED	TURBULENT		BACK	HIGH		
B	VOICED	CLOSED		FRONT	MID		CLOSED
D	VOICED	CLOSED		CENTRAL	HIGH	TIP MOVEMENT	
G	VOICED	CLOSED		BACK	HIGH		
HH	UNVOICED	TURBULENT		BACK	HIGH		
F	UNVOICED	TURBULENT		FRONT	MID		
TH	UNVOICED	TURBULENT		FRONT	HIGH	TIP MOVEMENT	
S	UNVOICED	TURBULENT		CENTRAL	HIGH		
SH	UNVOICED	TURBULENT		CENTRAL	MID		
V	VOICED	TURBULENT		FRONT	MID		
DH	UNVOICED	TURBULENT		FRONT	HIGH	TIP MOVEMENT	
Z	VOICED	TURBULENT		CENTRAL	HIGH		
ZH	VOICED	TURBULENT		CENTRAL	MID		
CH	UNVOICED	TURBULENT		CENTRAL	HIGH	TIP MOVEMENT	
JH	VOICED	TURBULENT		CENTRAL	HIGH	TIP MOVEMENT	
WH	UNVOICED	TURBULENT		BACK	MID		
EL	VOICED	OPEN		FRONT	HIGH	TIP MOVEMENT	
EM	VOICED	OPEN	NASAL	CENTRAL	MID		CLOSED
EN	VOICED	OPEN	NASAL	CENTRAL	MID	TIP MOVEMENT	
DX	VOICED	CLOSED		CENTRAL	MID	TIP MOVEMENT	
Q	UNVOICED	TURBULENT		CENTRAL	MID		
-	UNVOICED	CLOSED		CENTRAL	MID		CLOSED
+	VOICED	OPEN		CENTRAL	MID		

B.3. Influence Coefficients

Voiced	Unvoiced	4.0
Voiced	Nasalized	0.2
Unvoiced	Nasalized	6.0
Open	Closed	8.5
Open	Turbulent	7.0
Closed	Turbulent	4.0
Nasalized	Non-nasalized	2.5
Front	Central	1.0
Front	Back	1.0
Central	Back	1.0
Low	Middle	1.0
Low	High	1.5
Middle	High	1 0
Tip movement	No movement	0.4
Rounded	Normal	0.2
Rounded	Normal	0.2
Closed	Normal	0.3

Note: These coefficients are somewhat ad hoc and are likely to change over the next few years. Anyone wishing their current values should contact the author.

B.4. Phonetic Ambiguity Matrix - Theoretical Model

	-	P	B	T	D	K	G	F	V	DH	S	SH	HH	M	N	NX
-	100															
P	22	100														
B	22	5	100													
T	22	38	5	100												
D	22	5	38	5	100											
K	22	34	5	63	5	100										
G	22	5	34	5	63	5	100									
F	22	89	5	43	5	38	5	100								
V	5	22	22	22	22	22	22	22	100							
DH	22	56	5	68	5	52	5	63	22	100						
S	22	41	5	93	5	68	5	46	22	63	100					
SH	22	61	5	63	5	46	5	68	22	43	68	100				
HH	22	34	5	63	5	100	5	38	22	52	68	46	100			
M	10	2	32	2	12	2	11	2	18	2	2	2	2	100		
N	10	2	12	2	32	2	20	2	18	2	2	2	2	38	100	
NX	10	2	11	2	11	2	18	2	18	2	2	2	2	34	35	100
W	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
L	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
Y	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
UW	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
UH	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
OW	1	1	4	1	4	1	4	1	7	1	1	[illegible]	1	3	3	3
AO	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
AA	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
AH	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
ER	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
AE	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
EH	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
IH	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
IY	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3
AX	1	1	4	1	4	1	4	1	7	1	1	1	1	3	3	3

	W	L	Y	UW	UH	OW	AO	AA	AH	ER	AE	EH	IH	IY	AX
-															
P															
B															
T															
D															
K															
G															
F															
V															
DH															
S															
SH															
HH															
M															
N															
NX															
W	100														
L	86	100													
Y	63	63	100												
UW	43	43	58	100											
UH	43	43	58	100	100										
OW	63	63	40	68	68	100									
AO	43	43	33	56	56	68	100								
AA	43	43	33	56	56	68	100	100							
AH	93	93	58	46	46	68	46	46	100						
ER	86	100	63	43	43	63	43	43	93	100					
AE	43	43	33	32	32	38	56	56	46	43	100				
EH	63	63	40	38	38	56	38	38	68	63	68	100			
IH	43	43	58	56	56	38	32	32	46	43	56	68	100		
IY	43	43	58	56	56	38	32	32	46	43	56	68	100	100	
AX	93	93	58	46	46	68	46	46	100	93	46	68	46	46	100

Appendix C: TASK DEFINITIONS

This appendix contains descriptions of the languages analyzed in this thesis. Each description consists of a definition of the syntax of the language and a dictionary for it's vocabulary. Dictionaries give the allowed pronuncations for each word in the vocabulary. The first four tasks are not truely languages. but are sets of words we wished to analyze. They have been given a simple syntactic description which allows any word to follow any other word.

PHONS is a language consisting of a set of 33 phones. Describing the phones as a language makes possible the same analysis as for any other vocabulary. This means we can calculate the effective vocabulary size for the phones.

DIGIT: This vocabulary is the 10 digits. It was included because it was one of the first vocabularies used in speech recognition. It is still used, although usually for comparative purposes.

ALPHA: This vocabulary is the spoken letters of the alphabet. It is highly ambiguous phonetically and is therefore a good test case.

ADIG: Is the combination of the 10 digits and the 26 letters. Having this vocabulary allows one to evaluate the effect of combining two vocabularies.

CHESS: The original Hearsay I chess task language. It has a vocabulary of 25 words.

LIZ: Lizard is a small voice programming language with a vocabulary of 17 words. It has been used in the Harpy speech recognition system.

VP: This language is also a voice programming language. It has been used by both the Hearsay I system and the Harpy system. It is richer in it's syntax than Lizard and contains 37 words.

IBM: This is the IBM "New Raleigh" Language. It describes syntactically correct English-like sentences with little or no semantic interpretation.

LLBAS: A language developed by Lincoln Labs for use with their speech recognition system. It's task is displaying and controlling acoustic data. There are 236 words in it's vocabulary.

LLEXT: An "extended" version of LLBAS containing 410 words.

C.1. Phone Language

C.1.1 PHONS Syntax

```
<S>::=          [ <WORDS> ]

<WORDS>::=      <WORD> <WORDS>
                <WORD>

<WORD>::=       --
                P
                B
                T
                D
                K
                G
                F
                V
                DH
                S
                SH
                HH
                M
                N
                NX
                W
                L
                Y
                UW
                UH
                OW
                AO
                AA
                AH
                ER
                AE
                EH
                IH
                IY
                AX
                WH
```

C.1.2 PHONS Dictionary

-	-
P	P
B	B
T	T
D	D
K	K
G	G
F	F
V	V
DH	DH
S	S
SH	SH
HH	HH
M	M
N	N
NX	NX
W	W
L	L
Y	Y
UW	UW
UH	UH
OW	OW
AO	AO
AA	AA
AH	AH
ER	ER
AE	AE
EH	EH
IH	IH
IY	IY
AX	AX
WH	WH
←	←
[	-
]	-

C.2. Digit Language

C.2.1 DIGIT Syntax

<S>::=	[<WORDS>]
<WORDS>::=	<WORD> <WORDS>
	<WORD>
<WORD>::=	ZERO
	ONE
	TWO
	THREE
	FOUR
	FIVE
	SIX
	SEVEN
	EIGHT
	NINE

C.2.2 DIGIT Dictionary

ZERO	(-,0) S (-,0) (IH,IY) ER OW
ONE	(-,0) W AH N
TWO	(-,0) T (-,0) IH UW
THREE	(-,0) F (-,0) ER IY
FOUR	(-,0) F (-,0) AO ER
FIVE	(-,0) F (-,0) AA (AX,IH) V
SIX	(-,0) S (-,0) IH (-,0) K (-,0) S
SEVEN	(-,0) S (-,0) EH V (EH,AX) N
EIGHT	(-,0) EH (IH,AX) (-,0) T
NINE	(-,0) N AA IH N
[	-
]	-

C.3. Alphabet Language

C.3.1 ALPHA Syntax

<S>::=	[<WORDS>]
<WORDS>::=	<WORD> <WORDS>
	<WORD>
<WORD>::=	"A"
	"B"
	"C"
	"D"
	"E"
	"F"
	"G"
	"H"
	"I"
	"J"
	"K"
	"L"
	"M"
	"N"
	"O"
	"P"
	"Q"
	"R"
	"S"
	"T"
	"U"
	"V"
	"W"
	"X"
	"Y"
	"Z"

C.3.2 ALPHA Dictionary

"A"	(-,0) EH (IH,AX)
"B"	(-,0) B IY
"C"	(-,0) S IY
"D"	(-,0) D IY
"E"	(-,0) IY
"F"	(-,0) EH F
"G"	(-,0) G IY
"H"	(-,0) EH (IH,AX) (-,0) T SH
"I"	(-,0) AA IH
"J"	(-,0) D SH EH (IH,AX)
"K"	(-,0) K EH (IH,AX)
"L"	(-,0) EH L
"M"	(-,0) EH M
"N"	(-,0) EH N
"O"	(-,0) OW
"P"	(-,0) P IY
"Q"	(-,0) K Y UW
"R"	(-,0) AA ER
"S"	(-,0) EH S
"T"	(-,0) T IY
"U"	(-,0) Y UW
"V"	(-,0) V IY
"W"	(-,0) D AX B ((EH,0) L,0) Y UW
"X"	(-,0) EH K S
"Y"	(-,0) W AA IH
"Z"	(-,0) S IY
[	-
]	-

C.4. Alphabet-Digit Language

C.4.1 Alphabet-Digit Syntax

<S>::=	[<WORDS>]
<WORDS>::=	<WORD> <WORDS> <WORD>
<WORD>::=	"A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z" ZERO ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE

C.4.2 Alphabet-Digit Dictionary

"A"	(-,0) EH (IH,AX)
"B"	(-,0) B IY
"C"	(-,0) S IY
"D"	(-,0) D IY
"E"	(-,0) IY
"F"	(-,0) EH F
"G"	(-,0) G IY
"H"	(-,0) EH (IH,AX) (-,0) T SH
"I"	(-,0) AA IH
"J"	(-,0) D SH EH (IH,AX)
"K"	(-,0) K EH (IH,AX)
"L"	(-,0) EH L
"M"	(-,0) EH M
"N"	(-,0) EH N
"O"	(-,0) OW
"P"	(-,0) P IY
"Q"	(-,0) K Y UW
"R"	(-,0) AA ER
"S"	(-,0) EH S
"T"	(-,0) T IY
"U"	(-,0) Y UW
"V"	(-,0) V IY
"W"	(-,0) D AX B ((EH,0) L,0) Y UW
"X"	(-,0) EH K S
"Y"	(-,0) W AA IH
"Z"	(-,0) S IY
ZERO	(-,0) S (-,0) (IH,IY) ER OW
ONE	(-,0) W AH N
TWO	(-,0) T (-,0) IH UW
THREE	(-,0) F (-,0) ER IY
FOUR	(-,0) F (-,0) AO ER
FIVE	(-,0) F (-,0) AA (AX,IH) V
SIX	(-,0) S (-,0) IH (-,0) K (-,0) S
SEVEN	(-,0) S (-,0) EH V (EH,AX) N
EIGHT	(-,0) EH (IH,AX) (-,0) T
NINE	(-,0) N AA IH N
[	-
]	-

C.5. Chess language

C.5.1 Chess Syntax

```
<BIGMOVE>::=        [ <MOVE> ]
<MOVE>::=           <MOVE1><CHECK-WORD>
                    <MOVE1>

<MOVE1>::=          <REGULAR-MOVE>
                    <CAPTURE>
                    <CASTLE>

<CASTLE>::=         <CASTLE-WORD>ON<UNIROYAL>SIDE
                    <CASTLE-WORD><UNIROYAL>SIDE
                    <CASTLE-WORD>

<REGULAR-MOVE>::=   <PCE-LOC><MOVE-WORD><SQUARE>
                    <PAWN-LOC><MOVE-WORD><SQUARE38>

<CAPTURE>::=        <EP-PAWN><CAPTURE-WORD>PAWN EN-PASSENT
                    <PCE-LOC><CAPTURE-WORD><CMAN-LOC>
                    <PAWN-LOC><CAPTURE-WORD><PMAN-LOC>

<CASTLE-WORD>::=    CASTLE-S

<MOVE-WORD>::=      TO
                    MOVES-TO
                    GOES-TO

<CAPTURE-WORD>::=   TAKES
                    CAPTURES

<CHECK-WORD>::=     CHECK MATE
                    CHECK

<EP-PAWN>::=        <EP-PAWN-LOC>
                    <UNIROYAL><EP-PAWN-LOC>
                    <UNIROYAL><UNIPIECE><EP-PAWN-LOC>
                    <UNIPIECE><EP-PAWN-LOC>

<EP-PAWN-LOC>::=    PAWN ON <UNIROYAL><PIECE> FIVE
                    PAWN ON <NOPAWN> FIVE
                    PAWN

<CMAN-LOC>::=       <CPCE-LOC>
                    <PAWN-LOC>
```

```
<PCE-LOC>::=        <PCE-SPEC> ON <SQUARE>
                    <PCE-SPEC>

<PCE-SPEC>::=       <UNIROYAL><PIECE>
                    <NOPAWN>

<CPCE-LOC>::=       <CPCE-SPEC> ON <SQUARE>
                    <CPCE-SPEC>

<CPCE-SPEC>::=      <UNIROYAL><PIECE>
                    <NOPNOK>

<PAWN-LOC>::=       <PAWN-SPEC> ON <SQUARE27>
                    <PAWN-SPEC>

<PAWN-SPEC>::=      <UNIROYAL><UNIPIECE>PAWN
                    <UNIROYAL>PAWN
                    <UNIPIECE>PAWN
                    PAWN

<PMAN-LOC>::=       <CPCE-SPEC> ON <SQUARE38>
                    <CPCE-SPEC>
                    <PAWN-SPEC> ON <SQUARE37>
                    <PAWN-SPEC>

<SQUARE>::=         <UNIROYAL><PIECE><RANK>
                    <NOPAWN><RANK>

<SQUARE27>::=       <UNIROYAL><PIECE><RANK27>
                    <NOPAWN><RANK27>

<SQUARE38>::=       <UNIROYAL><PIECE><RANK38>
                    <NOPAWN><RANK38>

<SQUARE37>::=       <UNIROYAL><PIECE><RANK37>
                    <NOPAWN><RANK37>

<UNIROYAL>::=       KING-S
                    QUEEN-S

<UNIPIECE>::=       BISHOP-S
                    KNIGHT-S
                    ROOK-S

<NOPNOK>::=         QUEEN
                    BISHOP
                    KNIGHT
                    ROOK

<NOPAWN>::=         KING
```

```
                <NOPNOK>

<PIECE>::=      BISHOP
                KNIGHT
                ROOK

<RANK37>::=     THREE
                FOUR
                FIVE
                SIX
                SEVEN

<RANK27>::=     <RANK37>
                TWO

<RANK38>::=     <RANK37>
                EIGHT

<RANK>::=       <RANK38>
                ONE
                TWO
```

C.5.2 Chess Dictionary

BISHOP	(-,0) B (AX,IH) SH AX P
BISHOP-S	(-,0) B (AX,IH) SH AX P (S,0)
CAPTURES	(-,0) K AE P (-,0) T SH ER S
CASTLE-S	(-,0) K AE S (EH,0) L S
CHECK	(-,0) T SH EH K
EIGHT	(-,0) EH (IH,AX) T
EN-PASSENT	(-,0) AA N P AA S AA N
FIVE	(-,0) F AA IH V
FOUR	(-,0) F OW ER
GOES-TO	(-,0) G OW S T AX
KING	(-,0) K IH NX
KING-S	(-,0) K IH NX (S,0)
KNIGHT	(-,0) N AA IH T
KNIGHT-S	(-,0) N AA IH T (S,0)
MATE	(-,0) M EH (IH,AX) T
MOVES-TO	(-,0) M UW V S T AX
ON	(-,0) AA N
ONE	(-,0) W AH N
PAWN	(-,0) P AO N
QUEEN	(-,0) K W IY N
QUEEN-S	(-,0) K W IY N (S,0)
ROOK	(-,0) ER UH K
ROOK-S	(-,0) ER UH K (S,0)
SEVEN	(-,0) S EH V AX N
SIDE	(-,0) S AA IH D
SIX	(-,0) S IH K S
TAKES	(-,0) T EH (IH,AX) K S
THREE	(-,0) F ER IY
TO	(-,0) T AX
TWO	(-,0) T UW
[	-
]	-

C.6. Lizard Language

C.6.1 Lizard Syntax

```
<UTT>::=              [<COMMAND>]

<COMMAND>::=          <OP><SIGN-NUMBER>
                      DISPLAY

<OP>::=               ADD
                      SUBTRACT
                      MULTIPLY
                      DIVIDE
                      LOAD

<SIGN-NUMBEP>::=      MINUS <NUMBER>
                      <NUMBER>

<NUMBER>::=           <DIGIT>
                      <DIGIT><NUMBER-2>

<DIGIT>::=            ZERO
                      ONE
                      TWO
                      THREE
                      FOUR
                      FIVE
                      SIX
                      SEVEN
                      EIGHT
                      NINE

<NUMBER-2>::=         <DIGIT-2>
                      <DIGIT-2><NUMBER>

<DIGIT-2>::=          ZERO
                      ONE
                      TWO
                      THREE
                      FOUR
                      FIVE
                      SIX
                      SEVEN
                      EIGHT
                      NINE
```

C.6.2 Lizard Dictionary

ADD	(-,0) (HH,0) (AX,0) AE (- D,0)
DISPLAY	(-,0) D (IH,AX) S - P L EH (IH,0) (AX,0)
DIVIDE	(-,0) D (IH,AX) V (F,0) AH (IH,0)
EIGHT	(-,0) (HH,0) (AX,0) EH (- T,0)
FIVE	(-, (-,0)) F AH (IH,0) V
FOUR	(-, (-,0)) F AH ER
LOAD	(-,0) L OW (AX,0)
MINUS	(-,0) M AH (IH,0) (AX,0) N IH S
MULTIPLY	(-,0) M AA (EH,0) L (-,0) T AX ((-,0),-) P L AH (IH,0) (AX,0)
NINE	(-,0) N AH (IH,0) (AX,0) N
ONE	(-,0) W AH N
SEVEN	(-,0) S EH V (AX,AX,0) N
SIX	(-,0) S IH (,-, -) S
SUBTRACT	(-,0) S (AX,UH) - T ER AE (- T,0)
THREE	(-, (-,0)) F ER IY (AX,0)
TWO	(-, (-,0)) T IH UW
ZERO	(-,0) S (AX,0) IH (ER,0) OW
[	-
]	-

C.7. Voice Programming Language

C.7.1 Voice Programming Syntax

```
<REQUEST>::=      [ <COMMAND> ]

<COMMAND>::=      <SET-WORD> <SIMPLE-EXPRE> <IN-WORD> <VARIABLEDF>
                  <VARIABLE> <GET-WORD> <SIMPLE-EXPRF>
                  <SHOW-WORD> <SIMPLE-EXPRF>

<SET-WORD>::=     STORE
                  PUT

<IN-WORD>::=      IN
                  INTO

<GET-WORD>::=     GETS
                  BECOMES

<SHOW-WORD>::=    WHAT IS
                  SHOW

<BIN-OPE>::=      PLUS
                  MINUS
                  TIMES
                  DIVIDE
                  MOD
                  POWER
                  MAX
                  MIN

<UN-OPE>::=       NEGATE
                  ABSOLUTE
                  FACT

<BIN-OPF>::=      PLUS
                  MINUS
                  TIMES
                  DIVIDE
                  MOD
                  POWER
                  MAX
                  MIN

<UN-OPF>::=       NEGATE
                  ABSOLUTE
                  FACT
```

<SIMPLE-EXPRE>::=	<PRIMARYCE> <BIN-OPE> <PRIMARYDE> <UN-OPE> <PRIMARYDE> <PRIMARYDE>
<VARIABLECE>::=	ALPHA BETA GAMMA DELTA EPSILON
<PRIMARYCE>::=	<RADIXCE> <INTEGERCE> <INTEGERCE> <VARIABLECE>
<RADIXCE>::=	OCTAL DECIMAL
<INTEGERCE>::=	<DIGITACE> <INTEGERCE2> <DIGITACE>
<DIGITACE>::=	ZERO ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE
<INTEGERCE2>::=	<DIGITACE2><INTEGERCE> <DIGITACE2>
<DIGITACE2>::=	ZERO ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE
<VARIABLEDE>::=	ALPHA BETA GAMMA DELTA EPSILON

<VARIABLE>::= ALPHA
BETA
GAMMA
DELTA
EPSILON

<PRIMARYDE>::= <RADIXDE> <INTEGERDE>
<INTEGERDE>
<VARIABLEDE>

<RADIXDE>::= OCTAL
DECIMAL

<INTEGERDE>::= <DIGITADE> <INTEGERDE2>
<DIGITADE>

<DIGITADE>::= ZERO
ONE
TWO
THREE
FOUR
FIVE
SIX
SEVEN
EIGHT
NINE

<INTEGERDE2>::= <DIGITADE2><INTEGERDE>
<DIGITADE2>

<DIGITADE2>::= ZERO
ONE
TWO
THREE
FOUR
FIVE
SIX
SEVEN
EIGHT
NINE

<SIMPLE-EXPRF>::= <PRIMARYCF> <BIN-OPF> <PRIMARYDF>
<UN-OPF> <PRIMARYDF>
<PRIMARYDF>

<VARIABLECF>::= ALPHA
BETA
GAMMA
DELTA
EPSILON

```
<PRIMARYCF>::=      <RADIXCF> <INTEGERCF>
                    <INTEGERCF>
                    <VARIABLECF>

<RADIXCF>::=        OCTAL
                    DECIMAL

<INTEGERCF>::=      <DIGITACF> <INTEGERCF2>
                    <DIGITACF>

<DIGITACF>::=       ZERO
                    ONE
                    TWO
                    THREE
                    FOUR
                    FIVE
                    SIX
                    SEVEN
                    EIGHT
                    NINE

<INTEGERCF2>::=     <DIGITACF2><INTEGERCF>
                    <DIGITACF2>

<DIGITACF2>::=      ZERO
                    ONE
                    TWO
                    THREE
                    FOUR
                    FIVE
                    SIX
                    SEVEN
                    EIGHT
                    NINE

<VARIABLEDF>::=     ALPHA
                    BETA
                    GAMMA
                    DELTA
                    EPSILON

<PRIMARYDF>::=      <RADIXDF> <INTEGERDF>
                    <INTEGERDF>
                    <VARIABLEDF>

<RADIXDF>::=        OCTAL
                    DECIMAL

<INTEGERDF>::=      <DIGITADF> <INTEGERDF2>
                    <DIGITADF>
```

<DIGITADF>::=	ZERO
	ONE
	TWO
	THREE
	FOUR
	FIVE
	SIX
	SEVEN
	EIGHT
	NINE
<INTEGERDF2>::=	<DIGITADF2><INTEGERDF>
	<DIGITADF2>
<DIGITADF2>::=	ZERO
	ONE
	TWO
	THREE
	FOUR
	FIVE
	SIX
	SEVEN
	EIGHT
	NINE

C.7.2 Voice Programming Dictionary

ABSOLUTE	(-,0) (HH,0) (AX,0) AE (,-, -) S (AX,0) L UW (- T,0)
ALPHA	(-,0) (HH,0) AX AE (EH,0) L ((-,0),0) (F,0) (AH)
BECOMES	(-, (-,0)) (B,HH) (IY,IH) (,-, -) K AH M S
BETA	(-, (-,0)) (B,HH) EH (D,) (T,0) AH
DECIMAL	(-,0) D EH S M (EH,0) L
DELTA	(-,0) D EH L ((,N) ((-,0) T,0),D) AH
DIVIDE	(-,0) D AX V (-,0) Y (AX,0) ((AX,HH,0),0)
EIGHT	(-,0) (HH,0) (AX,0) EH (- T,0)
EPSILON	(-,0) (HH,0) (AX,0) (EH,AX) (-,, -) S (AX,0) L AO N
FACT	(-, (-,0),0) F AE (,-) (- T,0)
FIVE	(-, (-,0),0) F Y V
FOUR	(-, (-,0),0) F AH ER
GAMMA	(-,0) G AE M AH
GETS	(-,0) G IH (AX,0) (,-, -) S
IN	(-,0) (HH,0) (AX,IH) N
INTO	(-,0) (HH,0) (AX,IH) N (,0) (-,0) T AX
IS	(-,0) (HH,0) (AX,0) IH (IY,AX,0) (S (S,0),(S,0) S)
MAX	(-,0) M AE (,0) - S
MIN	(-,0) M IH N
MINUS	(-,0) M Y N AX S
MOD	(-,0) M AA
NEGATE	(-,0) N (AX,EH) (-,0) G EH (- T,0)
NINE	(-,0) N Y (AX,0) N
OCTAL	(-,0) AA (,0) - T (EH,0) L
ONE	(-,0) W AH N
PLUS	(-, (-,0)) P L AH S
POWER	(-, (-,0)) P AA UH ER
PUT	(-, (-,0)) P UH (- T,0)
SEVEN	(-,0) S EH V (AX,0) N
SHOW	(-,0) SH AH OW (OW (,0),0)
SIX	(-,0) S IH (,-, -) S
STORE	(-,0) S - T AH ER
THREE	(-, (-,0)) F (,0) ER IY
TIMES	(-, (-,0)) T Y M S
TWO	(-, (-,0)) T IH UW
WHAT	(-,0) (HH,0) W AA (- T,0)
ZERO	(-,0) S IH ER OW (AX,0)
(	-
)	-

C.8. IBM "New Raleigh" Language

C.8.1 IBM "New Raleigh" Syntax

```
<S> ::=        <BOX0> <BOX0X>
<BOX0X> ::=    <BOX1> <BOX1X>
               <BOX2> <BOX2X>
               <BOX3> <BOX3X>
               <BOX4> <BOX4X>
<BOX1X> ::=    <BOX5> <BOX5X>
               <BOX9> <BOX9X>
<BOX5X> ::=    <BOX9> <BOX9X>
<BOX9X> ::=    <BOX13> <BOX13X>
               <BOX14> <BOX14X>
<BOX13X> ::=   <BOX21> <BOX21X>
<BOX14X> ::=   <BOX24> <BOX24X>
               <BOX25> <BOX25X>
<BOX2X> ::=    <BOX6> <BOX6X>
               <BOX10> <BOX10X>
<BOX6X> ::=    <BOX10> <BOX10X>
<BOX10X> ::=   <BOX15> <BOX15X>
               <BOX16> <BOX16X>
<BOX15X> ::=   <BOX21> <BOX21X>
<BOX16X> ::=   <BOX24> <BOX24X>
               <BOX25> <BOX25X>
<BOX3X> ::=    <BOX7> <BOX7X>
               <BOX11> <BOX11X>
<BOX7X> ::=    <BOX11> <BOX11X>
<BOX11X> ::=   <BOX17> <BOX17X>
               <BOX18> <BOX18X>
<BOX17X> ::=   <BOX21> <BOX21X>
<BOX18X> ::=   <BOX24> <BOX24X>
               <BOX25> <BOX25X>
<BOX4X> ::=    <BOX8> <BOX8X>
               <BOX12> <BOX12X>
<BOX8X> ::=    <BOX12> <BOX12X>
<BOX12X> ::=   <BOX19> <BOX19X>
               <BOX20> <BOX20X>
<BOX19X> ::=   <BOX21> <BOX21X>
<BOX20X> ::=   <BOX24> <BOX24X>
               <BOX25> <BOX25X>
<BOX21X> ::=   <BOX22> <BOX22X>
               <BOX23> <BOX23X>
<BOX22X> ::=   <BOX28> <BOX28X>
               <BOX29> <BOX29X>
<BOX23X> ::=   <BOX26> <BOX26X>
               <BOX27> <BOX27X>
```

```
<BOX24X> ::=  <BOX26> <BOX26X>
              <BOX27> <BOX27X>
<BOX25X> ::=  <BOX28> <BOX28X>
              <BOX29> <BOX29X>
<BOX26X> ::=  <BOX30> <BOX30X>
<BOX30X> ::=  <BOX34> <BOX34X>
<BOX27X> ::=  <BOX31> <BOX31X>
<BOX31X> ::=  <BOX35> <BOX35X>
<BOX28X> ::=  <BOX32> <BOX32X>
<BOX32X> ::=  <BOX36> <BOX36X>
<BOX29X> ::=  <BOX33> <BOX33X>
<BOX33X> ::=  <BOX37> <BOX37X>
<BOX34X> ::=  <BOX38>
<BOX35X> ::=  <BOX38>
<BOX36X> ::=  <BOX38>
<BOX37X> ::=  <BOX38>
<BOX0> ::=    [
<BOX1> ::=    ONE
<BOX2> ::=    EACH
<BOX3> ::=    SOME
<BOX4> ::=    SHOULD
<BOX5> ::=    BAD
              BLACK
              GENTLE
              GREAT
              PRIMARY
              PROFICIENT
              QUIET
              RECOGNITION
              SMALL
              SUFFICIENT
<BOX6> ::=    DISTANT
              EAGER
              KIND
              LARGE
              NEW
              OTHER
              TINY
              TIRED
              TRUE
              UGLY
<BOX7> ::=    ACTIVE
              DEMOCRATIC
              FAIR
              LITTLE
              PRACTICAL
              POOR
              REAL
              SAFE
              SHORT
```

STRONG
<BOX8> ::= BACKWARD
BIG
CLOSE
GOOD
IMPORTANT
OLD
PASSIVE
RUGGED
SEPARATE
USELESS
<BOX9> ::= CONDITION
DURATION
GENERAL
PRIVATE
SERGEANT
TRAIN
VILLAGE
<BOX10> ::= DIVISION
PART
PERIOD
POWER
TIME
TOWN
WAR
<BOX11> ::= MATTERS
MEN
PEOPLE
PRACTICES
STREETS
TREATIES
WORKERS
<BOX12> ::= ACTIONS
BASES
BATTLES
COMMANDS
FORMS
GROUNDS
PLACES
<BOX13> ::= CONSIDERED
CREATED
GAVE
LIKED
MADE
MOVED
PERMITTED
WANTED
<BOX14> ::= CHANGES
DOES
FIGHTS

FEELS
GOES
LIVES
PROPOSES
VOTES
<BOX15> ::= CONTRIBUTED
CRITICIZED
DISTURBED
FORGOT
GOVERNED
HAD
SHOWED
TOOK
<BOX16> ::= APPEARS
APPROVES
DRINKS
HAS
IS
LOOKS
TAKES
WORKS
<BOX17> ::= ACCEPTED
APPLIED
BROUGHT
DETECTED
FOUND
OUTLAWED
REJECTED
SAVED
<BOX18> ::= ASK
GET
KNOW
MAKE
PAY
RAN
SURVIVE
WERE
<BOX19> ::= BE
CALL
CARRY
CONTROL
HAVE
THINK
TRY
TURN
<BOX20> ::= BELIEVE
COME
DO
DIRECT
FOLLOW

```
             PROCEED
             SEEM
             STAND
<BOX21> ::=  THE
<BOX22> ::=  BUILDING
             CAPTAIN
             CAUSE
             CITY
             COUNTRY
             LETTER
             MAJOR
             MAN
             NATION
             OFFICER
             REPORT
             THOUGHT
<BOX23> ::=  BUS
             CAMPAIGN
             FOOD
             GUN
             MOTION
             NAME
             RADIO
             SHIP
             STATE
             TELEPHONE
             THING
             WEAPON
<BOX24> ::=  AGAIN
             EXCESSIVELY
             LEAST
             MAJORLY
             MERELY
             MOSTLY
             NOT
             ONLY
             PRINCIPALLY
             PROPERLY
             SOMETIMES
             TRULY
<BOX25> ::=  ALWAYS
             FINALLY
             FREQUENTLY
             LESS
             MORE
             NEVER
             OCCASIONALLY
             OFTEN
             ONCE
             RARELY
```

```
                SELDOMLY
                USUALLY
<BOX26> ::=     ACROSS
                AT
                FROM
                ON
                TOWARD
                UNDER
<BOX27> ::=     AGAINST
                FOR
                IN
                INTO
                THROUGH
                TO
<BOX28> ::=     AROUND
                BEFORE
                DURING
                OVER
                PAST
                WITH
<BOX29> ::=     ABOUT
                AFTER
                AMONG
                BETWEEN
                BY
                WITHOUT
<BOX30> ::=     THOSE
<BOX31> ::=     THE
<BOX32> ::=     THE
<BOX33> ::=     THOSE
<BOX34> ::=     APPROACHES
                ENGINEERS
                GIRLS
                ISSUES
                LOCATIONS
                OPERATIONS
                PLANS
                PROBLEMS
                SITES
                ZONES
<BOX35> ::=     AIRPLANE
                BUSINESS
                ENGINE
                MACHINE
                MISSILE
                MOMENT
                ORDER
                PRODUCT
                USE
                YEAR
```

<BOX35> ::= CAPITOL
CONCERN
COVER
DAY
INTERVAL
LIFE
PURPOSE
SACRIFICE
VEHICLE
WEEK
<BOX37> ::= CAMPS
FIELDS
HOUSES
INTERESTS
METHODS
SCIENTISTS
SERVICES
SOLDIERS
SYSTEMS
TECHNIQUES
<BOX38> ::=]

C.8.2 IBM "New Raleigh" Dictionary

ABOUT	(-,0) AH B AA AX T
ACCEPTED	(-,0) IH K S EH P T (-,0) IH D
ACROSS	(-,0) AH K ER AA UH S
ACTIONS	(-,0) AE K SH AH N (-,0) S
ACTIVE	(-,0) AE K T IH V
AFTER	(-,0) AE F T ER
AGAIN	(-,0) AH G EH N
AGAINST	(-,0) AH G EH N S T
AIRPLANE	(-,0) EH AX ER (-,0) P L EH (IH,AX) N
ALWAYS	(-,0) AA UH L W EH (IH,AX) S
AMONG	(-,0) AH M AH NX
APPEARS	(-,0) AH P EH (IH,AX) AX ER (-,0) S
APPLIED	(-,0) AH P L AA AX (-,0) D
APPROACHES	(-,0) AH P ER OW (-,0) T SH (-,0) IH S
APPROVES	(-,0) AH P ER UW V (-,0) S
AROUND	(-,0) AH ER AA AX N D
ASK	(-,0) AE S K
AT	(-,0) AE T
BACKWARD	(-,0) B AE K W ER D
BAD	(-,0) B AE D
BASES	(-,0) B EH (IH,AX) S (-,0) IH S
BATTLES	(-,0) B AE T AH L (-,0) S
BE	(-,0) B EH (IH,AX)
BEFORE	(-,0) B EH (IH,AX) F OW AX ER
BELIEVE	(-,0) B EH (IH,AX) L EH (IH,AX) V
BETWEEN	(-,0) B EH (IH,AX) T W EH (IH,AX) N
BIG	(-,0) B IH G
BLACK	(-,0) B L AE K
BROUGHT	(-,0) B ER AA UH T
BUILDING	(-,0) B IH L D IH NX
BUS	(-,0) B AH S
BUSINESS	(-,0) B IH S N IH S
BY	(-,0) B AA AX
CALL	(-,0) K AA UH L
CAMPAIGN	(-,0) K AE M P EH (IH,AX) N
CAMPS	(-,0) K AE M P (-,0) S
CAPITOL	(-,0) K AE P IH T AH L
CAPTAIN	(-,0) K AE P T IH N
CARRY	(-,0) K AE ER EH (IH,AX)
CAUSE	(-,0) K AA UH S
CHANGES	(-,0) (-,0) T SH EH (IH,AX) N (-,0) D SH (-,0) IH S
CITY	(-,0) S IH T EH (IH,AX)
CLOSE	(-,0) K L OW S
COME	(-,0) K AH M
COMMANDS	(-,0) K AH M AE N D (-,0) S
CONCERN	(-,0) K AH N S ER N

CONDITION	(-,0) K AH N D IH SH AH N
CONSIDERED	(-,0) K AH N S IH D ER (-,0) D
CONTRIBUTED	(-,0) K AH N T ER IH B Y UW T (-,0) IH D
CONTROL	(-,0) K AH N T ER OW L
COUNTRY	(-,0) K AH N T ER EH (IH,AX)
COVER	(-,0) K AH V ER
CREATED	(-,0) K ER EH (IH,AX) EH (IH,AX) T (-,0) IH D
CRITICIZED	(-,0) K ER IH T IH S AA AX S (-,0) D
DAY	(-,0) D EH (IH,AX)
DEMOCRATIC	(-,0) D EH M AH K ER AE T IH K
DETECTED	(-,0) D EH (IH,AX) T EH K T (-,0) IH D
DIRECT	(-,0) D IH ER EH K T
DISTANT	(-,0) D IH S T AH N T
DISTURBED	(-,0) D IH S T ER B (-,0) D
DIVISION	(-,0) D IH V IH SH AH N
DO	(-,0) D UW
DOES	(-,0) D AH S
DRINKS	(-,0) D ER IH NX K (-,0) S
DURATION	(-,0) D Y UW ER EH (IH,AX) SH AH N
DURING	(-,0) D Y UW ER IH NX
EACH	(-,0) EH (IH,AX) (-,0) T SH
EAGER	(-,0) EH (IH,AX) G ER
ENGINE	(-,0) EH N (-,0) D SH IH N
ENGINEERS	(-,0) EH N (-,0) D SH IH N EH (IH,AX) AX ER (-,0) S
EXCESSIVELY	(-,0) EH K S EH S IH V (-,0) L EH (IH,AX)
FAIR	(-,0) F EH AX ER
FEELS	(-,0) F EH (IH,AX) L (-,0) S
FIELDS	(-,0) F EH (IH,AX) L D (-,0) S
FIGHTS	(-,0) F AA AX T (-,0) S
FINALLY	(-,0) F AA AX N AH L (-,0) EH (IH,AX)
FOLLOW	(-,0) F AA L OW
FOOD	(-,0) F UW D
FOR	(-,0) F AA UH AX ER
FORGOT	(-,0) F AA UH ER G AA T
FORMS	(-,0) F AA UH AX ER M (-,0) S
FOUND	(-,0) F AA AX N D
FREQUENTLY	(-,0) F ER EH (IH,AX) K W AH N T (-,0) L EH (IH,AX)
FROM	(-,0) F ER AH M
GAVE	(-,0) G EH (IH,AX) V
GENERAL	(-,0) (-,0) D SH EH N ER AH L
GENTLE	(-,0) (-,0) D SH EH N T AH L
GET	(-,0) G EH T
GIRLS	(-,0) G ER L (-,0) S
GOES	(-,0) G OW (-,0) S
GOOD	(-,0) G UH D
GOVERNED	(-,0) G AH V ER N (-,0) D
GREAT	(-,0) G ER EH (IH,AX) T
GROUNDS	(-,0) G ER AA AX N D (-,0) S
GUN	(-,0) G AH N
HAD	(-,0) HH AE D

HAS	(-,0) HH AE S
HAVE	(-,0) HH AE V
HOUSES	(-,0) HH AA AX S (-,0) IH S
IMPORTANT	(-,0) IH M P AA UH AX ER T AH N T
IN	(-,0) IH N
INTERESTS	(-,0) IH N T ER EH S T (-,0) S
INTERVAL	(-,0) IH N T ER V AH L
INTO	(-,0) IH N (-,0) T UW
IS	(-,0) IH S
ISSUES	(-,0) IH SH Y UW (-,0) S
KIND	(-,0) K AA AX N D
KNOW	(-,0) N OW
LARGE	(-,0) L AA AX ER (-,0) D SH
LEAST	(-,0) L EH (IH,AX) S T
LESS	(-,0) L EH S
LETTER	(-,0) L EH T ER
LIFE	(-,0) L AA AX F
LIKED	(-,0) L AA AX K (-,0) T
LITTLE	(-,0) L IH T AH L
LIVES	(-,0) L IH V (-,0) S
LOCATIONS	(-,0) L OW K EH (IH,AX) SH AH N (-,0) S
LOOKS	(-,0) L UH K (-,0) S
MACHINE	(-,0) M AH SH EH (IH,AX) N
MADE	(-,0) M EH (IH,AX) D
MAJOR	(-,0) M EH (IH,AX) (-,0) D SH ER
MAJORLY	(-,0) M EH (IH,AX) (-,0) D SH ER (-,0) L EH (IH,AX)
MAKE	(-,0) M EH (IH,AX) K
MAN	(-,0) M AE N
MATTERS	(-,0) M AE T ER (-,0) S
MEN	(-,0) M EH N
MERELY	(-,0) M EH (IH,AX) AX ER (-,0) L EH (IH,AX)
METHODS	(-,0) M EH F AH D (-,0) S
MISSILE	(-,0) M IH S IH L
MOMENT	(-,0) M OW M EH N T
MORE	(-,0) M OW AX ER
MOSTLY	(-,0) M OW S T (-,0) L EH (IH,AX)
MOTION	(-,0) M OW SH AH N
MOVED	(-,0) M UW V (-,0) D
NAME	(-,0) N EH (IH,AX) M
NATION	(-,0) N EH (IH,AX) SH AH N
NEVER	(-,0) N EH V ER
NEW	(-,0) N Y UW
NOT	(-,0) N AA T
OCCASIONALLY	(-,0) AH K EH (IH,AX) SH AH N AH L (-,0) EH (IH,AX)
OFFICER	(-,0) AA UH F IH S ER
OFTEN	(-,0) AA UH F AH N
OLD	(-,0) OW L D
ON	(-,0) AA N
ONCE	(-,0) W AH N S
ONE	(-,0) W AH N

ONLY	(-,0) OW N L EH (IH,AX)
OPERATIONS	(-,0) AA P AH ER EH (IH,AX) SH AH N (-,0) S
ORDER	(-,0) AA UH AX ER D ER
OTHER	(-,0) AH DH ER
OUTLAWED	(-,0) AA AX T L AA UH (-,0) D
OVER	(-,0) OW V ER
PART	(-,0) P AA AX ER T
PASSIVE	(-,0) P AE S IH V
PAST	(-,0) P AE S T
PAY	(-,0) P EH (IH,AX)
PEOPLE	(-,0) P EH (IH,AX) P AH L
PERIOD	(-,0) P EH (IH,AX) ER EH (IH,AX) IH D
PERMITTED	(-,0) P ER M IH T (-,0) IH D
PLACES	(-,0) P L EH (IH,AX) S (-,0) IH S
PLANS	(-,0) P L AE N (-,0) S
POOR	(-,0) P UW AX ER
POWER	(-,0) P AA AX AX ER
PRACTICAL	(-,0) P ER AE K T IH K AH L
PRACTICES	(-,0) P ER AE K T IH S (-,0) IH S
PRIMARY	(-,0) P ER AA AX M EH ER EH (IH,AX)
PRINCIPALLY	(-,0) P ER IH N S IH P (-,0) L EH (IH,AX)
PRIVATE	(-,0) P ER AA AX V IH T
PROBLEMS	(-,0) P ER AA B L AH M (-,0) S
PROCEED	(-,0) P ER OW S EH (IH,AX) D
PRODUCT	(-,0) P ER AA D AH K T
PROFICIENT	(-,0) P ER OW F IH SH AH N T
PROPERLY	(-,0) P ER AA P ER (-,0) L EH (IH,AX)
PROPOSES	(-,0) P ER OW P OW S (-,0) IH S
PURPOSE	(-,0) P ER P AH S
QUIET	(-,0) K W AA AX IH T
RADIO	(-,0) ER EH (IH,AX) D EH (IH,AX) OW
RAN	(-,0) ER AE N
RARELY	(-,0) ER EH AX ER (-,0) L EH (IH,AX)
REAL	(-,0) ER EH (IH,AX) L
RECOGNITION	(-,0) ER EH K IH G N IH SH AH N
REJECTED	(-,0) ER EH (IH,AX) (-,0) D SH EH K T (-,0) IH D
REPORT	(-,0) ER EH (IH,AX) P OW AX ER T
RUGGED	(-,0) ER AH G IH D
SACRIFICE	(-,0) S AE K ER IH F AA AX S
SAFE	(-,0) S EH (IH,AX) F
SAVED	(-,0) S EH (IH,AX) V (-,0) D
SCIENTISTS	(-,0) S AA AX IH N T IH S T (-,0) S
SEEM	(-,0) S EH (IH,AX) M
SELDOMLY	(-,0) S EH L D AH M (-,0) L EH (IH,AX)
SEPARATE	(-,0) S EH P ER IH T
SERGEANT	(-,0) S AA AX ER (-,0) D SH AH N T
SERVICES	(-,0) S ER V IH S (-,0) IH S
SHIP	(-,0) SH IH P
SHORT	(-,0) SH AA UH AX ER T
SHOULD	(-,0) SH UH D

SHOWED	(-,0) SH OW (-,0) D
SITES	(-,0) S AA AX T (-,0) S
SMALL	(-,0) S M AA UH L
SOLDIERS	(-,0) S OW L (-,0) D SH ER (-,0) S
SOME	(-,0) S AH M
SOMETIMES	(-,0) S AH M T AA AX M S
STAND	(-,0) S T AE N D
STATE	(-,0) S T EH (IH,AX) T
STREETS	(-,0) S T ER EH (IH,AX) T (-,0) S
STRONG	(-,0) S T ER AA UH NX
SUFFICIENT	(-,0) S AH F IH SH AH N T
SURVIVE	(-,0) S ER V AA AX V
SYSTEMS	(-,0) S IH S T IH M (-,0) S
TAKES	(-,0) T EH (IH,AX) K (-,0) S
TECHNIQUES	(-,0) T EH K N EH (IH,AX) K (-,0) S
TELEPHONE	(-,0) T EH L IH F OW N
THE	(-,0) DH AH
THING	(-,0) F IH NX
THINK	(-,0) F IH NX K
THOSE	(-,0) DH OW S
THOUGHT	(-,0) F AA UH T
THROUGH	(-,0) F ER IH AX
TIME	(-,0) T AA AX M
TINY	(-,0) T AA AX N EH (IH,AX)
TIRED	(-,0) T AA AX AX ER (-,0) D
TO	(-,0) T UW
TOOK	(-,0) T UH K
TOWARD	(-,0) T AH W AA UH AX ER D
TOWN	(-,0) T AA AX N
TRAIN	(-,0) T ER EH (IH,AX) N
TREATIES	(-,0) T ER EH (IH,AX) T EH (IH,AX) (-,0) S
TRUE	(-,0) T ER IH AX
TRULY	(-,0) T ER IH AX (-,0) L EH (IH,AX)
TRY	(-,0) T ER AA AX
TURN	(-,0) T ER N
UGLY	(-,0) AH G L EH (IH,AX)
UNDER	(-,0) AH N D ER
USE	(-,0) Y UW S
USELESS	(-,0) Y UW S (-,0) L EH S
USUALLY	(-,0) Y UW SH UW AH L (-,0) EH (IH,AX)
VEHICLE	(-,0) V EH (IH,AX) HH IH K AH L
VILLAGE	(-,0) V IH L IH (-,0) D SH
VOTES	(-,0) V OW T (-,0) S
WANTED	(-,0) W AA N [illegible] (-,0) IH D
WAR	(-,0) W AA UH AX ER
WEAPON	(-,0) W EH P AH N
WEEK	(-,0) W EH (IH,AX) K
WERE	(-,0) W ER
WITH	(-,0) W IH DH
WITHOUT	(-,0) W IH DH AA AX T

WORKERS	(-,0) W ER K (-,0) ER (-,0) S
WORKS	(-,0) W ER K (-,0) S
YEAR	(-,0) Y EH (IH,AX) AX ER
ZONES	(-,0) S OW N (-,0) S
[	-
]	-

C.9. LLBAS: Lincoln Lab "Basic" Language

C.9.1 LLBAS: Lincoln Lab "Basic" Syntax

```
<SENT> ::=        [ <SS> ]
<SS> ::=          <DIS>
                  <CON>
                  <CLR>
                  <GO>
                  <DEL>
                  <SK>
                  <MOV>
                  <COMP>
                  <GET>
                  <PIC>
                  <WRITE>
                  <PUT>
                  <LIST>
                  <OUTP>
                  <SET>


<DIS> ::=         <DISPV> <DISOBJ>
                  <DISPV> <DISOBJ> <DISWH>
<DISPV> ::=       DISPLAY
                  REDISPLAY
                  SHOW-ME
<DISOBJ> ::=      THE <DISOBJ1>
                  ALL MATCHES
                  ALL MATCHES <DW>
<DISOBJ1> ::=     <DISCLS>
                  <DISCLS> <DUDW>
                  FORMANTS
                  FORMANTS <DU>
                  FORMANT <PAR>
                  FORMANT <PAR> <DU>
<DUDW> ::=        <DU>
                  <DW>
<DW> ::=          OF <DET> <UTT>
<DET> ::=         THE
                  THIS
<DU> ::=          FOR THE <DISWRD>
<DISCLS> ::=      <PAR>
                  <MEAS> <PAR>
                  <LABS> LABELS
                  <DATFOR>
<MEAS> ::=        AVERAGE
```

```
                MAXIMUM
                MINIMUM
                TOTAL
<PAR> ::=       <PAR1>
                FIRST MOMENT
<PAR1> ::=      AMPLITUDE
                PITCH
                FREQUENCY
                GRAPH
                ENERGY
                ZEROCROSSING-DENSITY
<LABS> ::=      EDITED
                PHONEMIC
                HAND
<DATFOR> ::=    <DATFOR1>
                CONFUSION MATRIX
                EVENT ARRAY <S>
<DATFOR1> ::=   ENVELOPE <S>
                SPECTROGRAM <S>
                WAVEFORM <S>
                FORMANTS
                SPECTRUM
                SPECTRA
                SEGMENTATION
<DISWRD> ::=    <PHONS>
                <DISMOD> <PHONS>
                <DISMOD> WORD
<DISMOD> ::=    <LEN>
                <ORD>
<PHONS> ::=     <VOW>
                <POS> <VOW>
                <STOP>
                <VOIC> <STOP>
                <NAS>
                <FRIC>
                <VOIC> <FRIC>
                SONORANT <S>
                CONSONANT <S>
                DIPHTHONG <S>
<VOIC> ::=      VOICED
                UNVOICED
                VOICELESS
<POS> ::=       FRONT
                BACK
                HIGH
                LOW
                MID
<FRIC> ::=      FRICATIVE <S>
                AFFRICATE <S>
<STOP> ::=      STOP <S>
```

	PLOSIVE <S>
<VOW> ::=	VOWEL <S>
<NAS> ::=	NASAL <S>
	LIQUID <S>
	GLIDE <S>
<LEN> ::=	LONGEST
	SHORTEST
<ORD> ::=	FIRST
	SECOND
	THIRD
	FOURTH
<DISWH> ::=	ON THE <SCO>
<SCO> ::=	<DISDEV>
	<SCTYPE> <DISDEV>
<SCTYPE> ::=	HUGHES
	REFRESH
<DISDEV> ::=	SCOPE
	DISPLAY
	SCREEN
<UTT> ::=	ENTRY <S>
	UTTERANCE <S>
	SENTENCE <S>
	SLOT <S>
	FILE <S>

<CON> ::=	<CONV> <UNIT> <DIGIT> TO <COND>
	<CONV> TAPE <UNIT> <DIGIT> TO <COND>
<CONV> ::=	CONNECT
	ASSIGN
<COND> ::=	<DETM> <TERM>
	<TERM> <DIGIT>
	<TERM> NUMBER <DIGIT>
<DETM> ::=	THE
	THIS
	MY
<TERM> ::=	CONSOLE
	TERMINAL
<DIGIT> ::=	ONE
	TWO
	THREE
	FOUR
	FIVE
	SIX
	SEVEN
	EIGHT
	NINE
<UNIT> ::=	UNIT

```
<CLR> ::=        <CLRV> THE <CLRO>
<CLRV> ::=       CLEAR
                 ERASE
<CLRO> ::=       <SCO>
                 <PLOT> OF THE <DATFOR>
<PLOT> ::=       PLOT
                 GRAPH <S>
                 DISPLAY


<GO> ::=         <GOV> THE <MODE> MODE
<GOV> ::=        GO-INTO
                 SWITCH-TO
<MODE> ::=       SEARCH
                 GRAPHICS
                 DISPLAY


<DEL> ::=        <DELV> <DELO>
<DELV> ::=       DROP
                 DELETE
<DELO> ::=       <QUANT> <DATFOR>
                 <QUANT> <DATFOR> <DELOD>
                 <QUANT> <LABS> LABELS
                 <QUANT> <LABS> LABELS <DELOD>
                 THOSE <SPAN> <CARD> <TIME>
<DELOD> ::=      FROM THE <DATDEV>
<QUANT> ::=      <QUANT1>
                 ALL
                 ALL THE
<QUANT1> ::=     THE
                 THIS
<SPAN> ::=       <SPAN1>
                 LONGER THAN
                 GREATER THAN
                 SHORTER THAN
                 LESS THAN
<SPAN1> ::=      OVER
                 UNDER
<CARD> ::=       <DIGIT>
                 <DIGIT> <HUNDREDS>
                 <TENS>
                 <TENS> <DIGIT>
                 <TEENS>
<TENS> ::=       TWENTY
                 THIRTY
                 FORTY
                 FIFTY
                 SIXTY
                 SEVENTY
```

```
                    EIGHTY
                    NINETY
<TEENS> ::=         TEN
                    ELEVEN
                    TWELVE
                    THIRTEEN
                    FOURTEEN
                    FIFTEEN
                    SIXTEEN
                    SEVENTEEN
                    EIGHTEEN
                    NINETEEN
<HUNDREDS> ::=      HUNDRED
                    HUNDRED <DIGIT>
<TIME> ::=          SECONDS
                    MILLISECONDS


<SK> ::=            <SKV> <SKVO>
<SKV> ::=           SKIP
                    SKIP-OVER
<SKVO> ::=          THE <SEQ> <UTT>
                    THE <SEQ> <UTT> <SKVT>
                    TO THE <SEQ> <UTT>
                    TO THE <SEQ> <UTT> <SKVT>
<SKVT> ::=          ON UNIT <DIGIT>
                    ON TAPE UNIT <DIGIT>
<SEQ> ::=           <SEQ1>
                    <ORD>
<SEQ1> ::=          NEXT
                    CURRENT
                    INITIAL
                    LAST


<MOV> ::=           <MOVE> <MOVO>
<MOVE> ::=          MOVE
<MOVO> ::=          THE <MOVO2>
                    UNIT <CARD> <MOVO3>
                    TAPE UNIT <CARD> <MOVO3>
<MOVO2> ::=         <MARK> TO THE <SEQ> <SEG>
                    <MARK> TO THE <SEQ> <VOIC> <SEG>
                    <SIDE> <MARK> TO THE <SEQ> <SEG>
                    <SIDE> <MARK> TO THE <SEQ> <VOIC> <SEG>
                    <MARK> <DIR> <CARD> <TIME>
                    <SIDE> <MARK> <DIR> <CARD> <TIME>
                    TAPE <DIR> <CARD> <UTT>
                    TAPE TO <SEQ> <UTT>
<MOVO3> ::=         TO THE <SEQ> <UTT>
                    <DIR> <CARD> <UTT>
```

<SIDE> ::=	RIGHT
	LEFT
<MARK> ::=	BOUNDARY
	CURSOR
<SEG> ::=	FRAME
	SEGMENT
<DIR> ::=	FORWARD
	BACKWARD

<COMP> ::=	<COMPV> THE <COMPO>
<COMPV> ::=	COMPUTE
	CALCULATE
	RECOMPUTE
	RECALCULATE
<COMPO> ::=	<MEAS> <PAR>
	<MEAS> <PAR> <COMP2>
	DISTRIBUTION OF THE <PHONS>
	EFFECT OF <SHAPE> THE <LEV> TO <CARD>
<COMP2> ::=	IN <DET> <COMP3>
	IN <DET> <SEQ> <COMP3>
<COMP3> ::=	<PHONS>
	<SEG>
	<VOIC> <SEG>
<SHAPE> ::=	PUTTING
	SETTING
	INCREASING
	REDUCING
<LEV> ::=	THRESHOLD
	LEVEL
	GAIN

<GET> ::=	<GETV> <GETO>
<GETV> ::=	<GETV1>
	SEARCH FOR
<GETV1> ::=	FIND
	GET
	RETRIEVE
	GET-ME
	GIVE-ME
	TRY-TO-FIND
<GETO> ::=	<QUANT> <GETO1>
	THE <GETO2>
	<UTT> <CARD>
<GETO1> ::=	<UTT> INFORMATION
	<DISCLS> FROM THE <DATDEV>
<GETO2> ::=	RANGE OF THE <ORD> FORMANT
	RANGE OF THE <PAR>
	<UTT> <START> WITH <COM>

```
              <TENS> KILOHERTZ WAVEFORM <S>
              <SEQ> <UTT> <PREPIN> <DATDEV>
              <PHONS>
              <PHONS> <INSEN>
              <SEQ> <PHONS>
              <SEQ> <PHONS> <INSEN>
              <SEG>
              <SEG> <INSEN>
              <VOIC> <SEG>
              <VOIC> <SEG> <INSEN>
              <SEQ> <SEG>
              <SEQ> <SEG> <INSEN>
              <SEQ> <VOIC> <SEG>
              <SEQ> <VOIC> <SEG> <INSEN>
              <PHONS> FOR <GET2>
<PREPIN> ::=  ON
              IN THE
<INSEN> ::=   IN <GET2>
<GET2> ::=    <UTT> <CARD>
              <UTT>
              <SEQ> <UTT>
              THE <UTT>
              THE <SEQ> <UTT>
<DATDEV> ::=  <DATDEV1>
              DATA BASE
<DATDEV1> ::= TAPE
              DRUM
              DISK
              COMPUTER
<START> ::=   BEGINNING
              STARTING
<COM> ::=     RETRIEVE
              DELETE
              DISPLAY
              REDISPLAY


<PIC> ::=     <PICKV> <QUANT> <PHONS> <PICO>
<PICKV> ::=   PICKOUT
              SELECT
<PICO> ::=    IN THE <SEQ> <UTT>
              WITH THE <LIM> ENERGY
              ONLY FROM <UTT> LIST <CARD>
              WITH <ORDER> STRESS
<LIM> ::=     LEAST
              MOST
              HIGHEST
              LOWEST
<ORDER> ::=   PRIMARY
              SECONDARY
```

TERTIARY

```
<WRITE> ::=          <WRITV> <WRITO>
                     <WRITV> <WRITO> <WRITD>
<WRITV> ::=          WRITE
                     STORE
                     SAVE
<WRITO> ::=          <QUANT> <WRIT2>
                     EVERYTHING
                     THE <SEQ> <UTT>
<WRITD> ::=          ONTO <DATDEV1>
                     IN THE <DATDEV>
<WRIT2> ::=          <DISCLS>
                     FORMANT <PAR>
                     <COMPA> VALUE <S>
                     <COMPA> FIELD <S>
                     <TENS> KILOHERTZ WAVEFORM
<COMPA> ::=          COMPUTED
                     RECOMPUTED
                     CALCULATED
                     RECALCULATED


<PUT> ::=            <PUTV> <PUTO>
<PUTV> ::=           PUT
<PUTO> ::=           <WRITO>
                     <WRITO> <WRITD>
                     THE <SIDE> <MARK> ON THE <ORD> <SEG>


<LIST> ::=           <LISTV> <QUANT> <PHONS>
                     <LISTV> <QUANT> <PHONS> <LISTO>
<LISTV> ::=          LIST
                     PRINT
<LISTO> ::=          ON THE <PRDEV>
                     FROM <UTT> <CARD>
<PRDEV> ::=          XEROX
                     SCOPE


<OUTP> ::=           <OUTPUT> THE <OUTPUTO>
<OUTPUT> ::=         OUTPUT
<OUTPUTO> ::=        VECTOR OF <UTT> NAMES
                     <MEAS> ENERGY IN THE BAND


<SET> ::=            <SETV> THE <SETO>
<SETV> ::=           SET
                     RESET
```

```
<SETO> ::=      BATCH <TAG> TO <CARD>
                DEFAULT SPEAKER TO <ID>
                DEFAULT FOR SEX TO <SEX>
                DEFAULT FOR SITE TO <SITE>
                COLUMN <DIM> TO <CARD>
                INCREMENT TO <CARD>
<ID> ::=        <INIT>
                <NAME>
<INIT> ::=      JA
                RW
                SM
                CW
<NAME> ::=      ALLEN
                WIESEN
                MCCANDLESS
                WEINSTEIN
<DIM> ::=       WIDTH
                HEIGHT
<SEX> ::=       MALE
                FEMALE
<SITE> ::=      LL
                BBN
                SRI
                SDC
                CMU
<TAG> ::=       CODE
                TAG

<S> ::=         S
```

C.9.2 LLBAS: Lincoln Lab "Basic" Dictionary

AFFRICATE	(-,0) AE F ER (IH ,0) K EH T
ALL	(-,0) AO (L ,0)
ALLEN	(-,0) AE L (EH ,0) N
AMPLITUDE	(-,0) AE M P L (IH ,0) T UW D
ARRAY	(-,0) (AH ,0) ER EH (IH,AX)
ASSIGN	(-,0) AH S AA IH N
AVERAGE	(-,0) AE V ER IH (-,0) D SH
BACK	(-,0) B AE K
BACKWARD	(-,0) B AE K W ER D
BAND	(-,0) B AE N D
BASE	(-,0) B EH (IH,AX) S
BATCH	(-,0) B AE (-,0) T SH
BBN	(-,0) B IY B IY EH N
BEGINNING	(-,0) B IH G IH N IH NX
BOUNDARY	(-,0) B AA UH N D (AX ,0) ER IY
CALCULATE	(-,0) K AE L K (Y ,0) (AX ,0) L EH (IH,AX) T
CALCULATED	(-,0) K AE L K (Y ,0) (AX ,0) L EH (IH,AX) T AX D
CLEAR	(-,0) K L IH ER
CMU	(-,0) S IY EH M Y UW
CODE	(-,0) K OW D
COLUMN	(-,0) K AA L (AH ,0) M
COMPUTE	(-,0) K (AH ,0) M P (Y ,0) UW T
COMPUTED	(-,0) K AX M P (Y ,0) UW T AX D
COMPUTER	(-,0) K AX M P (Y ,0) UW T ER
CONFUSION	(-,0) K AX N F Y UW SH AX N
CONNECT	(-,0) K (AH ,0) N EH K T
CONSOLE	(-,0) K AA N S (EH,0) L
CONSONANT	(-,0) K AA N S (AH ,0) N AH N T
CURRENT	(-,0) K AH ER (AX ,0) N T
CURSOR	(-,0) K ER S ER
CW	(-,0) S IY D AH B (EH,0) L Y UW
DATA	(-,0) D EH (IH,AX) D AH
DEFAULT	(-,0) D IH F AO (L ,0) T
DELETE	(-,0) D (AH ,0) L IY T
DIPHTHONG	(-,0) D IH F F AA NX
DISK	(-,0) D IH S K
DISPLAY	(-,0) D (IH ,0) S P L EH (IH,AX)
DISTRIBUTION	(-,0) D IH S T (ER ,0) B Y UW SH (AX ,0) N
DROP	(-,0) D (ER ,0) AA P
DRUM	(-,0) D (ER ,0) AH M
EDITED	(-,0) EH D (AX ,0) D EH D
EFFECT	(-,0) IY F EH K T
EIGHT	(-,0) EH (IH,AX) T
EIGHTEEN	(-,0) EH (IH,AX) T IY N
EIGHTY	(-,0) EH (IH,AX) D IY
ELEVEN	(-,0) IY L EH V AX N

ENERGY	(-,0) EH N ER (-,0) D SH IY
ENTRIES	(-,0) EH N (T ,0) ER IY S
ENTRY	(-,0) EH N (T ,0) ER IY
ENVELOPE	(-,0) AH N V (AX ,0) L OW P
ERASE	(-,0) IY ER EH (IH,AX) S
EVENT	(-,0) IY V EH N T
EVERYTHING	(-,0) EH V ER IY F IH NX
FEMALE	(-,0) F IY M EH (IH,AX) L
FIELD	(-,0) F IY L D
FIFTEEN	(-,0) F IH F T IY N
FIFTY	(-,0) F IH F T IY
FILE	(-,0) F AA IH L
FIND	(-,0) F Y N D
FIRST	(-,0) F ER S (T ,0)
FIVE	(-,0) F AA IH V
FOR	(-,0) F (ER ,0) (AO ,0)
FORMANT	(-,0) F AO (ER ,0) M AH N T
FORMANTS	(-,0) F AO (ER ,0) M AH N T S
FORTY	(-,0) F AO T IY
FORWARD	(-,0) F AO (ER ,0) W ER D
FOUR	(-,0) F AO
FOURTEEN	(-,0) F AO T IY N
FOURTH	(-,0) F AO F
FRAME	(-,0) F ER EH (IH,AX) M
FREQUENCY	(-,0) F ER IY K W EH N S IY
FRICATIVE	(-,0) F ER IH K (IH ,0) D IH V
FROM	(-,0) F (ER ,0) AH M
FRONT	(-,0) F ER AH N T
GAIN	(-,0) G EH (IH,AX) N
GET	(-,0) G EH T
GET-ME	(-,0) G EH (T ,0) M IY
GIVE-ME	(-,0) G IH (V ,0) M IY
GLIDE	(-,0) G L AA IH D
GO-INTO	(-,0) G OW (W ,0) IH N T UW
GRAPH	(-,0) G ER AE F
GRAPHICS	(-,0) G ER AE F IH K S
GREATER	(-,0) G (S ,0) ER EH (IH,AX) D ER
HAND	(-,0) HH AE N D
HEIGHT	(-,0) HH AA IH T
HIGH	(-,0) HH AA IH
HIGHEST	(-,0) HH AA IH S T
HUGHES	(-,0) HH Y UW S
HUNDRED	(-,0) HH AH N D ER IH D
IN	(-,0) IH N
INCREASING	(-,0) IH N K ER IY S IH NX
INCREMENT	(-,0) IH N K ER M EH N T
INFORMATION	(-,0) IH N F ER M EH (IH,AX) SH (AX ,0) N
INITIAL	(-,0) IH N IH SH (AX ,0) L
JA	(-,0) D SH EH (IH,AX) (D ,0) EH (IH,AX)
KILOHERTZ	(-,0) K (S ,0) IH L OW HH ER T S

LABELS	(-,0) L EH (IH,AX) B (EH,0) L S
LAST	(-,0) L AE S T
LEAST	(-,0) L IY S T
LEFT	(-,0) L EH F T
LESS	(-,0) L EH S
LEVEL	(-,0) L EH V (EH,0) L
LIQUID	(-,0) L IH K W IH D
LIST	(-,0) L IH S T
LL	(-,0) EH L EH L
LONGER	(-,0) L AO N G ER
LONGEST	(-,0) L AO N G IH S T
LOW	(-,0) L OW
LOWEST	(-,0) L OW (W ,0) IH S T
MALE	(-,0) M EH (IH,AX) L
MATCHES	(-,0) M AE (-,0) T SH AX S
MATRIX	(-,0) M EH (IH,AX) T ER IH K S
MAXIMUM	(-,0) M AE K S (AX ,0) M AH M
MCCANDLESS	(-,0) M IH K AE N D L IH S
MID	(-,0) M IH D
MILLISECONDS	(-,0) M AH L (AH ,0) S EH K (AX ,0) N (T ,0) S
MINIMUM	(-,0) M IH N (AX ,0) M AX M
MODE	(-,0) M OW D
MOMENT	(-,0) M OW M EH N T
MOST	(-,0) M OW S T
MOVE	(-,0) M UW V
MY	(-,0) M AA IH
NAMES	(-,0) N EH (IH,AX) M S
NASAL	(-,0) N EH (IH,AX) S (EH,0) L
NEXT	(-,0) N EH K S T
NINE	(-,0) N AA IH N
NINETEEN	(-,0) N AA IH N T IY N
NINETY	(-,0) N AA IH N D IY
NUMBER	(-,0) N AH M B ER
OF	(-,0) (AH ,0) V
ON	(-,0) (AA ,0)
ONE	(-,0) W AH N
ONLY	(-,0) OW N L IY
ONTO	(-,0) AA IY N T Y (UW ,0)
OUTPUT	(-,0) AA UH (T ,0) P UH T
OVER	(-,0) OW V ER
PHONEMIC	(-,0) F OW N IY M IH K
PICKOUT	(-,0) P IH K AA UH T
PITCH	(-,0) P IH (-,0) T SH
PLOSIVE	(-,0) P L OW S IH V
PLOT	(-,0) P L AA T
PRIMARY	(-,0) P ER AA IH M (AX ,0) (ER ,0) IY
PRINT	(-,0) P ER IH N T
PUT	(-,0) P UH T
PUTTING	(-,0) P UH D IH NX
RANGE	(-,0) ER EH (IH,AX) N (-,0) D SH

RECALCULATE	(-,0) ER IY K AE L K (Y ,0) (AX ,0) L EH (IH,AX) T
RECALCULATED	(-,0) ER IY K AE L K (Y ,0) (AX ,0) L EH (IH,AX) T AX D
RECOMPUTE	(-,0) ER IY K (AX ,0) M P (Y ,0) UW T
RECOMPUTED	(-,0) ER IY K (AX ,0) M P (Y ,0) UW T AX D
REDISPLAY	(-,0) ER IY D (IH ,0) S P L EH (IH,AX)
REDUCING	(-,0) ER IY D (Y ,0) UW S IH NX
REFRESH	(-,0) ER IY F ER EH SH
RESET	(-,0) ER IY S EH T
RETRIEVE	(-,0) ER IY T (S ,0) ER IY V
RIGHT	(-,0) ER AA IH T
RW	(-,0) AH ER D AH B (EH,0) L Y UW
SAVE	(-,0) S EH (IH,AX) V
SCOPE	(-,0) S K OW P
SCREEN	(-,0) S K ER IY N
SDC	(-,0) EH S D IY S IY
SEARCH	(-,0) S ER (-,0) T SH
SECOND	(-,0) S EH K AX N D
SECONDARY	(-,0) S EH K (AX ,0) N D EH (ER ,0) (IY ,0)
SECONDS	(-,0) S EH K AX N S
SEGMENT	(-,0) S EH G M EH N T
SEGMENTATION	(-,0) S EH G M (EH ,0) EH (IH,AX) N T EH (IH,AX) SH AX N
SELECT	(-,0) S (AH ,0) EH K T
SEMIVOWEL	(-,0) S EH M IY V AA UH L
SENTENCE	(-,0) S EH N (T AX N ,0) S
SET	(-,0) S EH T
SETTING	(-,0) S EH D IH NX
SEVEN	(-,0) S EH V EH N
SEVENTEEN	(-,0) S EH V (EH ,0) N T IY N
SEVENTY	(-,0) S EH V AX N D IY
SEX	(-,0) S EH K S
SHORTER	(-,0) SH AO (ER ,0) T ER
SHORTEST	(-,0) SH AO (ER ,0) T AX S T
SHOW-ME	(-,0) SH OW M IY
SITE	(-,0) S AA IH T
SIX	(-,0) S IH K S
SIXTEEN	(-,0) S IH K S T IY N
SIXTY	(-,0) S IH K S D IY
SKIP	(-,0) S K IH P (S ,0)
SKIP-OVER	(-,0) S K IH P OW V AH
SLOT	(-,0) S L AA T
SM	(-,0) EH S EH M
SONORANT	(-,0) S OW L N ER (AX ,0) N T
SPEAKER	(-,0) S P IY K ER
SPECTRA	(-,0) S P EH (K ,0) T ER
SPECTROGRAM	(-,0) S P EH (K ,0) T ER G ER AE M
SPECTRUM	(-,0) S P EH (K ,0) T ER AH M
SRI	(-,0) EH S AH ER AA IH
STARTING	(-,0) S T AA (ER ,0) D IH NX
STOP	(-,0) S T AA P
STORE	(-,0) S T AO

STRESS	(-,0) S T (ER ,0) EH S
SWITCH-TO	(-,0) S W IH (-,0) T SH T Y UW
TAG	(-,0) T AE G
TAPE	(-,0) T (S ,0) EH (IH,AX) P
TEN	(-,0) T EH N
TERMINAL	(-,0) T ER M (IH ,0) N (EH,0) L
TERTIARY	(-,0) T ER SH (AX ,0) ER IY
THAN	(-,0) DH AE N
THE	(-,0) DH (AH ,0)
THIRD	(-,0) F ER D
THIRTEEN	(-,0) F ER T IY N
THIRTY	(-,0) F ER D IY
THIS	(-,0) DH IH S
THOSE	(-,0) DH OW S
THREE	(-,0) F ER IY
THRESHOLD	(-,0) F ER EH SH (EH,0) L D
TO	(-,0) T Y (UW ,0)
TOTAL	(-,0) T OW D (EH,0) L
TRY-TO-FIND	(-,0) T ER AA IH T AH F AA IH N D
TWELVE	(-,0) T W EH L V
TWENTY	(-,0) T W EH N T IY
TWO	(-,0) T Y UW
UNDER	(-,0) AH N D ER
UNIT	(-,0) Y UW N IH T
UNVOICED	(-,0) AH N V AO IH S T
UTTERANCE	(-,0) AH D ER (EH ,0) N S
VALUE	(-,0) V (AE ,0) L Y UW
VECTOR	(-,0) V EH (K ,0) T ER
VOICED	(-,0) V AO IH S T
VOICELESS	(-,0) V AO IH S L IH S
VOWEL	(-,0) V AA UH L
WAVEFORM	(-,0) W EH (IH,AX) V F ER M
WEINSTEIN	(-,0) W AA IH N S T AA IH N
WIDTH	(-,0) W IH D F
WIESEN	(-,0) W IY S AX N
WITH	(-,0) W IH F
WORD	(-,0) W ER D
WRITE	(-,0) ER AA IH T
XEROX	(-,0) S IH ER AA K S
ZEROCROSSING-DENSITY	(-,0) S IH ER OW K ER AA S IH NX D EH N S IH D IY
[	-
]	-

C.10. LLEXT: Lincoln Lab's "Extended" Language

C.10.1 LLEXT: Lincoln Lab's "Extended" Syntax

<SENT> ::=	[<SEN>]
<SEN> ::=	<BEGG> <SS>
	<SS>
<BEGG> ::=	PLEASE
	NOW
	WELL NOW
	NOW PLEASE
<SS> ::=	<DIS>
	<CON>
	<CLR>
	<GO>
	<DEL>
	<SK>
	<MOV>
	<COMP>
	<GET>
	<PIC>
	<WRITE>
	<PUT>
	<LIST>
	<SET>
	<MODSEG>
	<MODDIS>
	<CHNG>
	<QUEST>
<DIS> ::=	<DISPV> <DISOBJ>
	<DISPV> <DISOBJ> <DISWH>
<DISPV> ::=	DISPLAY
	REDISPLAY
	SHOW-ME
	PUT-UP
	EDIT
	I-WANT-TO-SEE
	LETS-SEE
<DISOBJ> ::=	THE <DISOBJ1>
	ALL MATCHES
	ALL MATCHES <DW>
<DISOBJ1> ::=	<DISCLS>
	<DISCLS> <DUDW>
	<LABS> LABELS
	<LABS> LABELS <DU>

```
                FORMANTS
                FORMANTS <DU>
                FORMANT <PAR2>
                FORMANT <PAR2> <DU>
<DUDW> ::=      <DU>
                <DW>
<DW> ::=        OF <DET> <UTT>
<DET> ::=       THE
                THIS
<DU> ::=        FOR <DISWRD>
<DISCLS> ::=    <PAR>
                <MEAS> <PAR>
                <LABS> LABELS
                <DATFOR>
                <PLOT> OF THE <PAR>
<MEAS> ::=      AVERAGE
                MAXIMUM
                MINIMUM
                TOTAL
                MAXIMA
                MINIMA
                DISTRIBUTION-OF
                RANGE-OF
<PAR> ::=       <PAR2>
                FIRST MOMENT
                SPECTRAL SLICES
<PAR2> ::=      AMPLITUDE
                PITCH
                FREQUENCY
                ENERGY
                DURATION
<LABS> ::=      EDITED
                PHONEMIC
                HAND
<DATFOR> ::=    <DATFOR1>
                <DATFOR2>
                <DATFOR3>
<DATFOR3> ::=   ENVELOPE <S>
                SPECTROGRAM <S>
                <SPEC> SPECTROGRAM <S>
                WAVEFORM <S>
                SPECTRUM
                SPECTRA
                SEGMENTATION
<DATFOR2> ::=   CONFUSION-MATRIX
                EVENT-ARRAY <S>
                PARSE-TREE <S>
                MEAN-VALUES
                FORMANTS
<DATFOR1> ::=   ENTRY-INFORMATION
```

```
                ZEROCROSSING-DENSITY
                PHONEMIC-TRANSCRIPTION <S>
                LEXICAL-TRANSCRIPTION <S>
                ZEROCROSSINGS
                GAIN-TABLE <S>
<SPEC> ::=      HOMOMORPHIC
                PREDICTIVE-CODING
<DISWRD> ::=    THE <PHONS>
                THE <DISMOD> <PHONS>
                THE <DISMOD> WORD
                THOSE <PHONS> <SPAN> <CARD> <TIME>
                <PHONS> AND <PHONS>
<DISMOD> ::=    <LEN>
                <ORD>
<PHONS> ::=     <VOW>
                <POS> <VOW>
                <STOP>
                <VOIC> <STOP>
                <NAS>
                <FRIC>
                <VOIC> <FRIC>
                SONORANT <S>
                CONSONANT <S>
                <CONS> CONSONANT <S>
                DIPHTHONG <S>
                SILENCE
                TRANSITION
                VOICING
<VOIC> ::=      VOICED
                UNVOICED
                VOICELESS
<POS> ::=       FRONT
                BACK
                HIGH
                LOW
                MID
<FRIC> ::=      FRICATIVE <S>
                AFFRICATE <S>
<STOP> ::=      STOP <S>
                PLOSIVE <S>
                BURST <S>
                ASPIRATE <S>
<VOW> ::=       VOWEL <S>
                SEMIVOWEL <S>
<NAS> ::=       NASAL <S>
                LIQUID <S>
                GLIDE <S>
<CONS> ::=      GLOTTAL
                INTERVOCALIC
                LABIAL
```

<LEN> ::=	LONGEST
	SHORTEST
<ORD> ::=	FIRST
	SECOND
	THIRD
	FOURTH
	FIFTH
	SIXTH
	SEVENTH
	EIGHTH
	NINTH
	TENTH
<DISWH> ::=	ON THE <SCO>
<SCO> ::=	<DISDEV>
	<SCTYPE> <DISDEV>
	SCAN-CONVERTOR
<SCTYPE> ::=	HUGHES
	REFRESH
<DISDEV> ::=	SCOPE
	DISPLAY
	SCREEN
<UTT> ::=	ENTRY <S>
	UTTERANCE <S>
	SENTENCE <S>
	SLOT <S>
	FILE <S>
	DATA
	STATEMENT <S>
	SAMPLE <S>
	EXAMPLE <S>

<CON> ::=	<CONV> <UNIT> <DIGIT>
	<CONV> <UNIT> <DIGIT> <COND>
<CONV> ::=	CONNECT
	ASSIGN
	LOAD
	REWIND
<COND> ::=	TO <DETM> <TERM>
	TO <TERM> <DIGIT>
	TO <TERM> NUMBER <DIGIT>
<DETM> ::=	THE
	THIS
	MY
<TERM> ::=	CONSOLE
	TERMINAL
<DIGIT> ::=	ONE
	TWO
	THREE
	FOUR

	FIVE
	SIX
	SEVEN
	EIGHT
	NINE
<UNIT> ::=	TAPE-UNIT
	UNIT
<CLR> ::=	<CLRV> THE <CLRO>
<CLRV> ::=	CLEAR
	ERASE
	CLEAN
	INITIALIZE
	REINITIALIZE
	REDO
	REFRESH
<CLRO> ::=	<SCO>
	<PLOT> OF THE <PAR>
	<DATFOR>
	<PLOT> OF THE <DATFOR>
<PLOT> ::=	PLOT <S>
	GRAPH <S>
	FUNCTIONS
	LINE <S>
<GO> ::=	<GOV> THE <MODE> MODE
<GOV> ::=	GO-INTO
	SWITCH-TO
	SET-TO
<MODE> ::=	SEARCH
	GRAPHICS
	DISPLAY
	INPUT
<DEL> ::=	<DELV> <DELO>
<DELV> ::=	DROP
	DELETE
	FORGET
	REMOVE
	SCRATCH
	THROW-AWAY
<DELO> ::=	<QUANT> <DATFOR>
	<QUANT> <DATFOR> <DELOD>
	<QUANT> <LABS> LABELS
	<QUANT> <LABS> LABELS <DELOD>
	THOSE <SPAN> <CARD> <TIME>
<DELOD> ::=	FROM THE <DATDEV>

<QUANT> ::=	<QUANT1>
	ALL
	ALL THE
<QUANT1> ::=	THE
	THIS
<SPAN> ::=	<SPAN1>
	LONGER THAN
	GREATER THAN
	SHORTER THAN
	LESS THAN
	BETWEEN <CARD> AND
	LOWER THAN
	HIGHER THAN
<SPAN1> ::=	OVER
	UNDER
	ABOVE
	BELOW
<CARD> ::=	<DIGIT>
	<DIGIT> <HUNDREDS>
	<TENS>
	<TENS> <DIGIT>
	<TEENS>
<TENS> ::=	TWENTY
	THIRTY
	FORTY
	FIFTY
	SIXTY
	SEVENTY
	EIGHTY
	NINETY
<TEENS> ::=	TEN
	ELEVEN
	TWELVE
	THIRTEEN
	FOURTEEN
	FIFTEEN
	SIXTEEN
	SEVENTEEN
	EIGHTEEN
	NINETEEN
<HUNDREDS> ::=	HUNDRED
	HUNDRED <DIGIT>
	HUNDRED <TEENS>
	HUNDRED <TENS>
	HUNDRED <TENS> <DIGIT>
<TIME> ::=	SECONDS
	MILLISECONDS
<SK> ::=	<SKV> <SKVO>

```
<SKV> ::=      SKIP
               SKIP <DIR> TO
               SKIP-OVER
               SKIP-OVER TO
               MOVE TO
               MOVE <DIR> TO
               CONTINUE TO
               CONTINUE <DIR> TO
               SEARCH FOR
               SEARCH <DIR> FOR
               READ
               READ TO
               READ <DIR>
               READ <DIR> TO
               GO TO
               GO <DIR> TO
               PROCEED TO
               PROCEED <DIR> TO
               <SKV1>
<SKV1> ::=     SKIP-TO
               FIND
               RETRIEVE
               SHIFT-TO
               GET
               GET-ME
               GIVE-ME
               TRY-TO-FIND
               PICKOUT
               SELECT
               GO-ON-TO
               I-WANT-TO-SEE
               I-WANT-ONLY
               I-ONLY-WANT
               LET-ME-SEE
               LETS-SEE
               PARSE
               READ-IN
               RETURN-TO
<SKVO> ::=     <SEQ> <UTT>
               <SEQ> <UTT> <SKVT>
               <UTT> <CARD>
               <UTT> <CARD>  <SPKR>
<SKVT> ::=     ON UNIT <DIGIT>
               ON TAPE UNIT <DIGIT>
               FROM THE <DATDEV>
               <WITH> <DETA> <PHONSEG>
<SEQ> ::=      THE <SEQ1>
               THE <ORD>
               ANOTHER
               THE
```

```
<SEQ1> ::=     NEXT
               CURRENT
               INITIAL
               LAST
               FINAL
               BEGINNING
               ENDING
               BRIEF
               OTHER
               PREVIOUS
               PROBLEM
<SPKR> ::=     <BY> <NAME>
               <BY> <INIT>
               <BY> A <SEX> SPEAKER
               <BY> SPEAKER NUMBER <DIGIT>
<DETA> ::=     A
               THE
<BY> ::=       BY
               SPOKEN-BY


<MOV> ::=      <MOVE> <MOVO>
<MOVE> ::=     MOVE
               SHIFT
<MOVO> ::=     THE <MOVO2>
               UNIT <CARD> <MOVO3>
               TAPE UNIT <CARD> <MOVO3>
<MOVO2> ::=    <MARK> TO <SEQ> <SEG>
               <MARK> TO <SEQ> <VOIC> <SEG>
               <SIDE> <MARK> TO <SEQ> <SEG>
               <SIDE> <MARK> TO <SEQ> <VOIC> <SEG>
               <MARK> TO <SEQ> <PHONS> <SEG>
               <SIDE> <MARK> TO <SEQ> <PHONS> <SEG>
               <MARK> TO THE <PHONS> <SEG>
               <SIDE> <MARK> TO THE <PHONS> <SEG>
               <MARK> <DIR> <CARD> <TIME>
               <SIDE> <MARK> <DIR> <CARD> <TIME>
               TAPE <DIR> <CARD> <UTT>
               TAPE TO <SEQ> <UTT>
<MOVO3> ::=    TO THE <ORD> <UTT>
               <DIR> <CARD> <UTT>
<SIDE> ::=     RIGHT
               LEFT
               PREVIOUS
               NEXT
<MARK> ::=     BOUNDARY
               CURSOR
               MARKER <S>
               LABEL <S>
               DESCRIPTOR
```

	POINTER
	POINT
<SEG> ::=	FRAME <S>
	SEGMENT <S>
	EXAMPLE <S>
	EVENT <S>
	OCCURANCE <S>
	PHONEME <S>
	SECTION <S>
	PHRASE <S>
<DIR> ::=	FORWARD
	BACKWARD
	EARLIER
	LATER
	ALONG
	AHEAD
	BACK
<COMP> ::=	<COMPV> THE <COMPO>
<COMPV> ::=	COMPUTE
	CALCULATE
	RECOMPUTE
	RECALCULATE
	DO
	REDO
	AVERAGE
	NORMALIZE
<COMPO> ::=	<MEAS> <PAR>
	<MEAS> <PAR> <COMP2>
	<DATFOR2> <COMP4>
	DISTRIBUTION OF THE <PHONS>
	EFFECT OF <SHAPE> THE <LEV> TO <CARD>
<COMP2> ::=	IN <COMP3>
	IN <SEQ> <COMP3>
<COMP3> ::=	<PHONS>
	<SEG>
	<VOIC> <SEG>
<COMP4> ::=	FOR <DET> <UTT>
	<COMP2>
<SHAPE> ::=	PUTTING
	SETTING
	INCREASING
	REDUCING
<LEV> ::=	THRESHOLD
	LEVEL
	GAIN
	CUTOFF

<GET> ::=	<GETV> <GETO>
<GETV> ::=	<GETV1>
	SEARCH FOR
<GETV1> ::=	FIND
	GET
	RETRIEVE
	GET-ME
	GIVE-ME
	TRY-TO-FIND
	PICKOUT
	SELECT
	I-WANT-TO-SEE
	I-ONLY-WANT
	I-WANT-ONLY
	LET-ME-SEE
	LETS-SEE
<GETO> ::=	<QUANTG1> <GETO1>
	<GETO2>
	<SEQ> <SEG>
<GETO1> ::=	<UTT> INFORMATION
	<DISCLS> <WITH> A <PHONSEG>
	<DISCLS> FOR <GET2>
	<DISCLS> FROM THE <DATDEV>
<GETO2> ::=	THE RANGE OF THE <ORD> FORMANT
	THE RANGE OF THE <PAR>
	THE <UTT> <START> WITH <COM>
	THE <TENS> KILOHERTZ WAVEFORM <S>
	<SEQ> <UTT> <PREPIN> <DATDEV>
	THE <PHONS>
	THE <PHONS> ONLY <INSEN>
	<SEQ> <PHONS>
	<SEQ> <PHONS> <INSEN>
	<SEQ> <SEG>
	<SEQ> <SEG> <INSEN>
	<SEQ> <PHONS> <SEG>
	<SEQ> <PHONS> <SEG> <INSEN>
	THE <SEG>
	THE <SEG> <INSEN>
	THE <PHONS> <SEG>
	THE <PHONS> <SEG> <INSEN>
	THE <PHONS> FOR <GET2>
<PREPIN> ::=	ON
	IN-THE
<INSEN> ::=	IN <GET2>
	FROM <GET2>
<GET2> ::=	<UTT> <CARD>
	<UTT>
	<SEQ> <UTT>
	THE <UTT>
	THE <SEQ> <UTT>

	<QUANT2> <UTT> <SPKR>
	<UTT> <LISTG> <CARD>
<DATDEV> ::=	<DATDEV1>
	<DATDEV2>
<DATDEV1> ::=	TAPE
	DRUM
	DISK
<DATDEV2> ::=	DATA-BASE
	COMPUTER
<START> ::=	BEGINNING
	STARTING
<COM> ::=	RETRIEVE
	DELETE
	DISPLAY
	REDISPLAY
<QUANTG1> ::=	ALL
	ALL THE
	THE
<QUANT2> ::=	EACH
	EVERY
<LISTG> ::=	NUMBER
	LIST

<PIC> ::=	<PICKV> <QUANT> <PHONS> <PICO>
<PICKV> ::=	PICKOUT
	SELECT
	FIND
	LOCATE
	SHOW-ME
<PICO> ::=	IN <SEQ> <UTT>
	WITH THE <LIM> ENERGY
	ONLY FROM <UTT> LIST <CARD>
	WITH <ORDER> STRESS
<LIM> ::=	LEAST
	MOST
	HIGHEST
	LOWEST
<ORDER> ::=	PRIMARY
	SECONDARY
	TERTIARY

<WRITE> ::=	<WRITV> <WRITO>
	<WRITV> <WRITO> <WRITD>
<WRITV> ::=	WRITE
	STORE
	SAVE
	PUT
	KEEP

	INSERT
	ADD
<WRITO> ::=	<QUANT> <WRIT2>
	EVERYTHING
	<SEQ> <UTT>
<WRITD> ::=	ONTO <DATDEV1>
	ONTO THE <DATDEV1>
	IN THE <DATDEV2>
	ON THE <DISDEV>
	INTO THE <DATDEV2>
	ON <DATDEV1>
	ON THE <DATDEV1>
<WRIT2> ::=	<DISCLS>
	FORMANT <PAR>
	<COMPA> VALUE <S>
	<CHOICE> FROM <THIS> ANALYSIS
	<COMPA> FIELD <S>
	<TENS> KILOHERTZ WAVEFORM
<COMPA> ::=	COMPUTED
	RECOMPUTED
	CALCULATED
	RECALCULATED
	NORMALIZED
<CHOICE> ::=	CHOICE
	RESULT
	PROBABILITIES
	PERCENTAGES
<THIS> ::=	THIS
	THESE
<PUT> ::=	<PUTV> <PUTO>
<PUTV> ::=	PUT
	INSERT
	ADD
	POSITION
	MOVE
	SHIFT
	SLIDE
<PUTO> ::=	<QUANT> <DISCLS> <PUTWHERE>
	THE <SIDE> <MARK> ON THE <ORD> <SEG>
<PUTWHERE> ::=	<SPAN1> THE <DISCLS>
	HERE
	THERE
<LIST> ::=	<LISTV> <LISTWHT>
	<LISTV> <LISTWHT> <LISTO>
<LISTV> ::=	LIST
	PRINT

TYPE
OUTPUT
<LISTWHT> ::= <QUANT> <PHONS>
<QUANT> <DATFORI>
THE VECTOR OF <UTT> NAMES
THE <MEAS> ENERGY IN THE BAND
<LISTO> ::= ON THE <PRDEV>
FROM <UTT> <CARD>
<PRDEV> ::= XEROX
SCOPE
<WITH> ::= WITH
CONTAINING
PRECEEDING
FOLLOWING
FOLLOWED-BY
STARTING-WITH
ENDING-WITH
PRECEEDED-BY
<PHONSEG> ::= <PHONS> <SEG2>
<PHONS> <PHONS> <SEG2>
<PHONS> <PHONS> <PHONS> <SEG2>
<SEG2> ::= <SEG>
STRING
SEQUENCE
COMBINATION

<MODDIS> ::= <MODV> <MODO>
<MODV> ::= BOOST
DECREASE
DOUBLE
ENLARGE
INCREASE
REDUCE
SPREAD-OUT
<MODO> ::= THE <DISCLS>
FOR THE <DISWRD>

<MODSEG> ::= <MODSV> <MODSO>
<MODSV> ::= ABSORB
ADD
INSERT
POSITION
TAKE
<MODSO> ::= <DET> <PHONSEG> <ADV> <DET> <PHONS>
<ADV> ::= AFTER
BEFORE
PRECEEDING
FOLLOWING

```
                    AT

<SET> ::=           <SETV> THE <SETO>
<SETV> ::=          SET
                    RESET
<SETO> ::=          BATCH <TAG> TO <CARD>
                    DEFAULT SPEAKER TO <ID>
                    DEFAULT FOR SEX TO <SEX>
                    DEFAULT FOR SITE TO <SITE>
                    COLUMN <DIM> TO <CARD>
                    INCREMENT TO <CARD>
<ID> ::=            <INIT>
                    <NAME>
<INIT> ::=          JA
                    RW
                    SM
                    CW
<NAME> ::=          ALLEN
                    WIESEN
                    MCCANDLESS
                    WEINSTEIN
<DIM> ::=           WIDTH
                    HEIGHT
<SEX> ::=           MALE
                    FEMALE
<SITE> ::=          LL
                    BBN
                    SRI
                    SDC
                    CMU
<TAG> ::=           CODE
                    TAG


<CHNG> ::=          <CHNGV> <DET> <PHONSEG> A <PHONS>
                    CHANGE THE <PHONSEG> TO A <PHONS>
                    ASSIGN <PHONS> TO THE <PHONSEG>
                    COMPARE THE <PHSG> WITH THE <PHSG>
<CHNGV> ::=         NAME
                    DESIGNATE
                    LABEL
                    MARK
                    CALL
                    MAKE
<PHSG> ::=          <PHONSEG>
                    <PHONS>


<QUEST> ::=         WHO OWNS <UTTOWN>
```

	WHERE <AXIL> <EXIST>
	<QUESTV> <UTT> HAVE <DATFOR> <DEVWHR>
	WHAT IS THE <WHATS>
<QUESTV> ::=	HOW MANY
	WHAT
	WHICH
<AXIL> ::=	IS
	ARE
	WAS
<DEVWHR> ::=	ON <DATDEV1>
	IN <DATDEV2>
<UTTOWN> ::=	<UTT> <CARD>
	<UTT> <BY> <SPKR>
	<SEQ1> <UTT>
<EXIST> ::=	<DATFOR> FOR THE <UTTOWN>
	<UTTOWN>
<WHATS> ::=	<PAR2>
	<DATFOR2>
	<DATFOR1>
	<LABS> LABELS
	OWNER'S-NAME
<S>::=	S

C.10.2 LLEXT: Lincoln Lab's "Extended" Dictionary

A	(-,0) AE
ABOUT	(-,0) (AH ,0) B AA UH T
ABOVE	(-,0) (AH ,0) B AH V
ABSORB	(-,0) (AH ,0) B S OW (ER ,0) B
ADD	(-,0) AE D
AFFRICATE	(-,0) AE F ER (IH ,0) K EH T
AFTER	(-,0) AE F D ER
AHEAD	(-,0) (AH ,0) HH EH D
ALL	(-,0) AO (L ,0)
ALLEN	(-,0) AE L (EH ,0) N
ALONG	(-,0) (AH ,0) AO NX
AMPLITUDE	(-,0) AE M P L (IH ,0) T UW D
ANALYSIS	(-,0) AE N AE L IH S AX S
AND	(-,0) AE N
ANOTHER	(-,0) (AH ,0) N AH DH ER
ARE	(-,0) AO ER
ASPIRATE	(-,0) AE S P ER (AX ,0) T
ASSIGN	(-,0) AH S AA IH N
AT	(-,0) AE T
AVERAGE	(-,0) AE V ER IH (-,0) D SH
BACK	(-,0) B AE K
BACKWARD	(-,0) B AE K W ER D
BAND	(-,0) B AE N D
BATCH	(-,0) B AE (-,0) T SH
BBN	(-,0) B IY B IY EH N
BEFORE	(-,0) B IH F OW (ER ,0)
BEGINNING	(-,0) B IH G IH N IH NX
BELOW	(-,0) B IH L OW
BETWEEN	(-,0) B IH T W IY N
BOOST	(-,0) B UW S T
BOUNDARY	(-,0) B AA UH N D (AX ,0) ER IY
BRIEF	(-,0) B ER IY F
BURST	(-,0) B ER S T
BY	(-,0) B AA IH
CALCULATE	(-,0) K AE L K (Y ,0) (AX ,0) L EH (IH,AX) T
CALCULATED	(-,0) K AE L K (Y ,0) (AX ,0) L EH (IH,AX) T AX D
CALL	(-,0) K AO L
CHANGE	(-,0) T SH EH (IH,AX) N (-,0) D SH
CHOICE	(-,0) T SH AO IH S
CLEAN	(-,0) K L IY N
CLEAR	(-,0) K L IH ER
CMU	(-,0) S I[illegible] M Y UW
CODE	(-,0) K OW D
COLUMN	(-,0) K AA L (AH ,0) M
COMBINATION	(-,0) K AO M B (IH ,0) N EH (IH,AX) SH AX N
COMPARE	(-,0) K AA L M P EH (ER ,0)

COMPUTE	(-,0) K (AH ,0) M P (Y ,0) UW T
COMPUTED	(-,0) K AX M P (Y ,0) UW T AX D
COMPUTER	(-,0) K AX M P (Y ,0) UW T ER
CONFUSION-MATRIX	(-,0) K AX N F Y UW SH AX N EH (IH,AX) T ER IH K S
CONNECT	(-,0) K (AH ,0) N EH K T
CONSOLE	(-,0) K AA N S (EH,0) L
CONSONANT	(-,0) K AA N S (AH ,0) N AH N T
CONTAINING	(-,0) K AX N T EH (IH,AX) N IH NX
CONTINUE	(-,0) K AX N T IH N Y UH
CPS	(-,0) S IY P IY EH S
CURRENT	(-,0) K AH ER (AX ,0) N T
CURSOR	(-,0) K ER S ER
CUTOFF	(-,0) K AH D AO F
CW	(-,0) S IY D AH B (EH,0) L Y UW
CYCLES-PER-SECOND	(-,0) S AA IH K (EH,0) L S P ER S EH K AX N D
DATA	(-,0) D EH (IH,AX) D AH
DATA-BASE	(-,0) D EH (IH,AX) D AH B EH (IH,AX) S
DECREASE	(-,0) D (IY ,0) K ER IY S
DEFAULT	(-,0) D IH F AO (L ,0) T
DELETE	(-,0) D (AH ,0) L IY T
DESCRIPTOR	(-,0) D (IH ,0) S K ER IH P T ER
DESIGNATE	(-,0) D EH S IH G N EH (IH,AX) T
DIPHTHONG	(-,0) D IH F F AA NX
DISK	(-,0) D IH S K
DISPLAY	(-,0) D (IH ,0) S P L EH (IH,AX)
DISTRIBUTION	(-,0) D IH S T (ER ,0) B Y UW SH (AX ,0) N
DISTRIBUTION-OF	(-,0) D IH S T (ER ,0) B Y UW SH (AX ,0) N (AH ,0) V
DO	(-,0) D UW
DOUBLE	(-,0) D AH B (EH,0) L
DROP	(-,0) D (ER ,0) AA P
DRUM	(-,0) D (ER ,0) AH M
DURATION	(-,0) D ER EH (IH,AX) SH (AH ,0) N
EACH	(-,0) IY (-,0) T SH
EARLIER	(-,0) ER L IY ER
EDIT	(-,0) EH D IH T
EDITED	(-,0) EH D (AX ,0) D EH D
EFFECT	(-,0) IY F EH K T
EIGHT	(-,0) EH (IH,AX) T
EIGHTEEN	(-,0) EH (IH,AX) T IY N
EIGHTH	(-,0) EH (IH,AX) F
EIGHTY	(-,0) EH (IH,AX) D IY
ELEVEN	(-,0) IY L EH V AX N
END	(-,0) EH N D
ENDING	(-,0) EH N D IH NX
ENDING WITH	(-,0) EH N D IH NX W IH F
ENERGY	(-,0) EH N ER (-,0) D SH IY
ENLARGE	(-,0) EH N L AO (ER ,0) (-,0) D SH
ENTRY INFORMATION	

	(-,0) EH N (T ,0) ER IY IH N F ER M EH (IH,AX) SH (AX ,0) N
ENTRY	(-,0) EH N (T ,0) ER IY
ENVELOPE	(-,0) AH N V (AX ,0) L OW P
ERASE	(-,0) IY ER EH (IH,AX) S
EVENT-ARRAY	(-,0) IY V EH N T (AH ,0) ER EH (IH,AX)
EVENT	(-,0) IY V EH N T
EVERY	(-,0) EH V ER IY
EVERYTHING	(-,0) EH V ER IY F IH NX
EXAMPLE	(-,0) EH G S AE M P (EH,0) L
FEMALE	(-,0) F IY M EH (IH,AX) L
FIELD	(-,0) F IY L D
FIFTEEN	(-,0) F IH F T IY N
FIFTH	(-,0) F IH F F
FIFTY	(-,0) F IH F T IY
FILE	(-,0) F AA IH L
FINAL	(-,0) F AA IH N (EH,0) L
FIND	(-,0) F Y N D
FIRST	(-,0) F ER S (T ,0)
FIT	(-,0) F IH T
FIVE	(-,0) F AA IH V
FOLLOWED-BY	(-,0) F AO L OW B AA IH
FOLLOWING	(-,0) F AO L OW (W ,0) (IH ,0) NX
FOR	(-,0) F (ER ,0) (AO ,0)
FORGET	(-,0) F ER G EH T
FORMANT	(-,0) F AO (ER ,0) M AH N T
FORMANTS	(-,0) F AO (ER ,0) M AH N T S
FORTY	(-,0) F AO T IY
FORWARD	(-,0) F AO (ER ,0) W ER D
FOUR	(-,0) F AO
FOURTEEN	(-,0) F AO T IY N
FOURTH	(-,0) F AO F
FRAME	(-,0) F ER EH (IH,AX) M
FREQUENCIES	(-,0) F ER IY K W EH N S IY S
FREQUENCY	(-,0) F ER IY K W EH N S IY
FRICATIVE	(-,0) F ER IH K (IH ,0) D IH V
FROM	(-,0) F (ER ,0) AH M
FRONT	(-,0) F ER AH N T
FUNCTIONS	(-,0) F AH N K SH (AH ,0) N
GAIN	(-,0) G EH (IH,AX) N
GAIN-TABLE	(-,0) G EH (IH,AX) N T EH (IH,AX) B (EH,0) L
GET	(-,0) G EH T
GET-ME	(-,0) G EH (T ,0) M IY
GIVE-ME	(-,0) G IH (V ,0) M IY
GLIDE	(-,0) G L AA IH D
GLOTTAL	(-,0) G L AA D (EH,0) L
GO	(-,0) G OW
GO-INTO	(-,0) G OW (W ,0) IH N T UW
GO-ON-TO	(-,0) G OW AA N T UW
GRAPH	(-,0) G ER AE F
GRAPHICS	(-,0) G ER AE F IH K S

GREATER	(-,0) G (S ,0) ER EH (IH,AX) D ER
HALF	(-,0) HH AE
HAND	(-,0) HH AE N D
HAVE	(-,0) HH AE V
HEADER	(-,0) HH EH D ER
HEIGHT	(-,0) HH AA IH T
HERE	(-,0) HH IY (ER ,0)
HIGH	(-,0) HH AA IH
HIGHER	(-,0) HH AA IH ER
HIGHEST	(-,0) HH AA IH S T
HOMOMORPHIC	(-,0) HH OW M (OW ,0) M OW (ER ,0) F IH K
HOW	(-,0) HH AA UH
HUGHES	(-,0) HH Y UW S
HUNDRED	(-,0) HH AH N D ER IH D
I-ONLY-WANT	(-,0) AA IH OW N L IY W AA N T
I-WANT-ONLY	(-,0) AA IH W AA N T OW N L IY
I-WANT-TO-SEE	(-,0) AA IH W AA N T UW S IY
IN	(-,0) IH N
IN-THE	(-,0) IH N DH (AH ,0)
INCREASE	(-,0) IH N K ER IY S
INCREASING	(-,0) IH N K ER IY S IH NX
INCREMENT	(-,0) IH N K ER M EH N T
INFORMATION	(-,0) IH N F ER M EH (IH,AX) SH (AX ,0) N
INITIAL	(-,0) IH N IH SH (AX ,0) L
INITIALIZE	(-,0) IH N IH SH (AX ,0) L AA IH S
INPUT	(-,0) IH N P UW T
INSERT	(-,0) IH N S ER T
INTERVOCALIC	(-,0) IY N T ER V OW K AE L IH K
INTO	(-,0) IH N T UW
IS	(-,0) IY S
JA	(-,0) D SH EH (IH,AX) (D ,0) EH (IH,AX)
KEEP	(-,0) K IY P
KILOHERTZ	(-,0) K (S ,0) IH L OW HH ER T S
LABEL	(-,0) L EH (IH,AX) B (EH,0) L
LABELS	(-,0) L EH (IH,AX) B (EH,0) L S
LABIAL	(-,0) L EH (IH,AX) B IY (EH,0) L
LAST	(-,0) L AE S T
LATER	(-,0) L EH (IH,AX) D ER
LEAST	(-,0) L IY S T
LEFT	(-,0) L EH F T
LESS	(-,0) L EH S
LET-ME-SEE	(-,0) L EH (IH,AX) T M IY S IY
LETS-SEE	(-,0) L EH (IH,AX) S IY
LEVEL	(-,0) L EH V (EH,0) L
LEXICAL-TRANSCRIPTION	
	(-,0) L EH K S (IH ,0) K (EH,0) L T ER AE N S K ER IH P SH (AH ,0) N
LINE	(-,0) L AA IH N
LIQUID	(-,0) L IH K W IH D
LIST	(-,0) L IH S T

LL	(-,0) EH L EH L
LOAD	(-,0) L OW D
LOCATE	(-,0) L OW K EH (IH,AX) T
LONGER	(-,0) L AO N G ER
LONGEST	(-,0) L AO N G IH S T
LOW	(-,0) L OW
LOWER	(-,0) L OW (W ,0) ER
LOWEST	(-,0) L OW (W ,0) IH S T
MAKE	(-,0) M EH (IH,AX) K
MALE	(-,0) M EH (IH,AX) L
MANY	(-,0) M EH N IY
MARK	(-,0) M AA (ER ,0) K
MARKER	(-,0) M AA (ER ,0) K ER
MATCHES	(-,0) M AE (-,0) T SH AX S
MAXIMA	(-,0) M AE K S (IH ,0) M AH
MAXIMUM	(-,0) M AE K S (AX ,0) M AH M
MCCANDLESS	(-,0) M IH K AE N D L IH S
MEAN-VALUES	(-,0) M IY N V (AE ,0) L Y UW (S ,0)
MID	(-,0) M IH D
MILLISECONDS	(-,0) M AH L (AH ,0) S EH K (AX ,0) N (T ,0) S
MINIMA	(-,0) M IH N (IH ,0) M AH
MINIMUM	(-,0) M IH N (AX ,0) M AX M
MODE	(-,0) M OW D
MOMENT	(-,0) M OW M EH N T
MOST	(-,0) M OW S T
MOVE	(-,0) M UW V
MY	(-,0) M AA IH
NAME	(-,0) N EH (IH,AX) M
NAMES	(-,0) N EH (IH,AX) M S
NASAL	(-,0) N EH (IH,AX) S (EH,0) L
NEXT	(-,0) N EH K S T
NINE	(-,0) N AA IH N
NINETEEN	(-,0) N AA IH N T IY N
NINETY	(-,0) N AA IH N D IY
NINTH	(-,0) N AA IH N F
NORMALIZE	(-,0) N OW (ER ,0) M (EH,0) L AA IH S
NORMALIZED	(-,0) N OW (ER ,0) M (EH,0) L AA IH S D
NOW	(-,0) N AA UH
NUMBER	(-,0) N AH M B ER
OCCURANCE	(-,0) (AH ,0) K ER EH N S (IH ,0)
OF	(-,0) (AH ,0) V
ON	(-,0) (AA ,0)
ONE	(-,0) W AH N
ONLY	(-,0) OW N L IY
ONTO	(-,0) AA IY N T Y (UW ,0)
OTHER	(-,0) AH DH ER
OUT	(-,0) AA UH T
OUTPUT	(-,0) AA UH (T ,0) P UH T
OVER	(-,0) OW V ER
OWNER'S-NAME	(-,0) OW N ER S N EH (IH,AX) M

OWNS	(-,0) OW N S
PARSE	(-,0) P AA (ER ,0) S
PARSE-TREE	(-,0) P AA (ER ,0) S T ER IY
PART	(-,0) P AA (ER ,0) T
PERCENTAGES	(-,0) P ER S EH N T EH (IH,AX) (-,0) D SH S
PHONEME	(-,0) F OW N IY M
PHONEMIC	(-,0) F OW N IY M IH K
PHONEMIC-TRANSCRIPTION	
	(-,0) F OW N IY M IH K T ER AE N S K ER IH P SH (AH ,0) N
PHRASE	(-,0) F ER EH (IH,AX) S (AX ,0)
PICKOUT	(-,0) P IH K AA UH T
PITCH	(-,0) P IH (-,0) T SH
PLEASE	(-,0) P L IY S
PLOSIVE	(-,0) P L OW S IH V
PLOT	(-,0) P L AA T
POINT	(-,0) P AO IH N T
POINTER	(-,0) P AO IH N T ER
POSITION	(-,0) P AH S IH SH (AH ,0) N
PRECEEDED-BY	(-,0) P ER IY S IY D AX D AA IH
PRECEEDING	(-,0) P ER IY S IY D IH NX
PREDICTIVE-CODING	
	(-,0) P ER IY D IH (K ,0) T IH (V ,0) K OW D AX NX
PREVIOUS	(-,0) P ER IY V Y AH S
PRIMARY	(-,0) P ER AA IH M (AX ,0) (ER ,0) IY
PRINT	(-,0) P ER IH N T
PROBABILITIES	(-,0) P ER AA B (AH ,0) B IH L IH D IY S
PROBLEM	(-,0) P ER AA B L AH M
PROCEED	(-,0) P ER OW S IY D
PUT	(-,0) P UH T
PUT-UP	(-,0) P UH T AH P
PUTTING	(-,0) P UH D IH NX
RANGE	(-,0) ER EH (IH,AX) N (-,0) D SH
RANGE-OF	(-,0) ER EH (IH,AX) N (-,0) D SH (AH ,0) V
READ	(-,0) ER IY D
READ-IN	(-,0) ER IY D IH N
RECALCULATE	(-,0) ER IY K AE L K (Y ,0) (AX ,0) L EH (IH,AX) T
RECALCULATED	(-,0) ER IY K AE L K (Y ,0) (AX ,0) L EH (IH,AX) T AX D
RECOMPUTE	(-,0) ER IY K (AX ,0) M P (Y ,0) UW T
RECOMPUTED	(-,0) ER IY K (AX ,0) M P (Y ,0) UW T AX D
REDISPLAY	(-,0) ER IY D (IH ,0) S P L EH (IH,AX)
REDO	(-,0) ER IY D UW
REDUCE	(-,0) ER IY D UW S
REDUCING	(-,0) ER IY D (Y ,0) UW S IH NX
REFRESH	(-,0) ER IY F ER EH SH
REINITIALIZE	(-,0) ER IY IH N IH SH (AX ,0) AA IH S
REMOVE	(-,0) ER IY M UW V
RESET	(-,0) ER IY S EH T
RESULT	(-,0) ER IY S AH L T
RETRIEVE	(-,0) ER IY T (S ,0) ER IY V
RETURN-TO	(-,0) ER IY T ER N T Y (UW ,0)

REWIND	(-,0) ER IY W AA IH N D
RIGHT	(-,0) ER AA IH T
RW	(-,0) AH ER D AH B (EH,0) L Y UW
SAMPLE	(-,0) S AE M P (EH,0) L
SAVE	(-,0) S EH (IH,AX) V
SCAN-CONVERTOR	(-,0) S K AE N K (AX ,0) N V ER D ER
SCOPE	(-,0) S K OW P
SCRATCH	(-,0) S K ER AE (-,0) T SH
SCREEN	(-,0) S K ER IY N
SDC	(-,0) EH S D IY S IY
SEARCH	(-,0) S ER (-,0) T SH
SECOND	(-,0) S EH K AX N D
SECONDARY	(-,0) S EH K (AX ,0) N D EH (ER ,0) (IY ,0)
SECONDS	(-,0) S EH K AX N S
SECTION	(-,0) S EH (IH,AX) K SH (AH ,0) N
SEGMENT	(-,0) S EH G M EH N T
SEGMENTATION	(-,0) S EH G M (EH ,0) EH (IH,AX) N T EH (IH,AX) SH AX N
SELECT	(-,0) S (AH ,0) EH K T
SEMIVOWEL	(-,0) S EH M IY V AA UH L
SENTENCE	(-,0) S EH N (T AX N ,0) S
SEQUENCE	(-,0) S IY K W EH N S
SET	(-,0) S EH T
SET-TO	(-,0) S EH T UW
SETTING	(-,0) S EH D IH NX
SEVEN	(-,0) S EH V EH N
SEVENTEEN	(-,0) S EH V (EH ,0) N T IY N
SEVENTH	(-,0) S EH V AX N F
SEVENTY	(-,0) S EH V AX N D IY
SEX	(-,0) S EH K S
SHIFT	(-,0) S IH F T
SHIFT-TO	(-,0) SH IH F T Y (UW ,0)
SHORTER	(-,0) SH AO (ER ,0) T ER
SHORTEST	(-,0) SH AO (ER ,0) T AX S T
SHOW-ME	(-,0) SH OW M IY
SILENCE	(-,0) S AA IH L (IH ,0) N S
SITE	(-,0) S AA IH T
SIX	(-,0) S IH K S
SIXTEEN	(-,0) S IH K S T IY N
SIXTH	(-,0) S IH K S F
SIXTY	(-,0) S IH K S D IY
SKIP	(-,0) S K IH P (S ,0)
SKIP-OVER	(-,0) S K IH P OW V AH
SKIP-TO	(-,0) S K IH T Y (UW ,0)
SLICES	(-,0) S L AA IH S AH S
SLIDE	(-,0) S L AA IH D
SLOT	(-,0) S L AA T
SM	(-,0) EH S EH M
SONORANT	(-,0) S OW L N ER (AX ,0) N T
SPEAKER	(-,0) S P IY K ER

SPECTRA	(-,0) S P EH (K ,0) T ER
SPECTRAL	(-,0) S P EH (K ,0) T (ER ,0) (EH,0) L
SPECTROGRAM	(-,0) S P EH (K ,0) T ER G ER AE M
SPECTRUM	(-,0) S P EH (K ,0) T ER AH M
SPOKEN-BY	(-,0) S P OW K (AH ,0) N B AA IH
SPREAD-OUT	(-,0) S P ER EH D AA UH T
SRI	(-,0) EH S AH ER AA IH
STARTING	(-,0) S T AA (ER ,0) D IH NX
STARTING-WITH	(-,0) S T AA (ER ,0) D IH NX W IH F
STATEMENT	(-,0) S T EH (IH,AX) (T ,0) M EH N (T ,0)
STOP	(-,0) S T AA P
STORE	(-,0) S T AO
STRESS	(-,0) S T (ER ,0) EH S
STRING	(-,0) S T (ER ,0) IH NX
SWITCH-TO	(-,0) S W IH (-,0) T SH T Y UW
TAG	(-,0) T AE G
TAKE	(-,0) T (S ,0) EH (IH,AX) K
TAPE	(-,0) T (S ,0) EH (IH,AX) P
TAPE-UNIT	(-,0) T (S ,0) EH (IH,AX) P IH N IH T
TEN	(-,0) T EH N
TENTH	(-,0) T EH N F
TERMINAL	(-,0) T ER M (IH ,0) N (EH,0) L
TERTIARY	(-,0) T ER SH (AX ,0) ER IY
THAN	(-,0) DH AE N
THAT	(-,0) DH AA T
THE	(-,0) DH (AH ,0)
THERE	(-,0) DH EH ER
THESE	(-,0) DH IY S
THIRD	(-,0) F ER D
THIRTEEN	(-,0) F ER T IY N
THIRTY	(-,0) F ER D IY
THIS	(-,0) DH IH S
THOSE	(-,0) DH OW S
THREE	(-,0) F ER IY
THRESHOLD	(-,0) F ER EH SH (EH,0) L D
THROW-AWAY	(-,0) F ER OW (W ,0) (AH ,0) (W ,0) EH (IH,AX)
TIME	(-,0) T AA IH M
TO	(-,0) T Y (UW ,0)
TOTAL	(-,0) T OW D (EH,0) L
TRANSITION	(-,0) T ER AE N S IH SH (AH ,0) N
TRY-TO-FIND	(-,0) T ER AA IH T AH F AA IH N D
TWELVE	(-,0) T W EH L V
TWENTY	(-,0) T W EH N T IY
TWO	(-,0) T Y UW
TYPE	(-,0) T AA IH P
UNDER	(-,0) AH N D ER
UNIT	(-,0) Y UW N IH T
UNVOICED	(-,0) AH N V AO IH S T
UTTERANCE	(-,0) AH D ER (EH ,0) N S
VALUE	(-,0) V (AE ,0) L Y UW

VECTOR	(-,0) V EH (K ,0) T ER
VOICED	(-,0) V AO IH S T
VOICELESS	(-,0) V AO IH S L IH S
VOICING	(-,0) V AO IH S IH NX
VOWEL	(-,0) V AA UH L
WAS	(-,0) W AH S
WAVEFORM	(-,0) W EH (IH,AX) V F ER M
WEINSTEIN	(-,0) W AA IH N S T AA IH N
WELL	(-,0) W EH L
WHAT	(-,0) HH W AA T
WHERE	(-,0) HH W EH ER
WHICH	(-,0) HH W IH (-,0) T SH
WHO	(-,0) HH UW
WIDTH	(-,0) W IH D F
WIESEN	(-,0) W IY S AX N
WILL	(-,0) W IH L
WITH	(-,0) W IH F
WORD	(-,0) W ER D
WRITE	(-,0) ER AA IH T
XEROX	(-,0) S IH ER AA K S
ZEROCROSSING-DENSITY	(-,0) S IH ER OW K ER AA S IH NX D EH N S IH D IY
ZEROCROSSINGS	(-,0) S IH ER OW K ER AA S IH NX S
[	-
]	-

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 18 AFOSR - TR - 77 - 0055	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 ANALYSIS OF LANGUAGES FOR MAN-MACHINE VOICE COMMUNICATION,		5. TYPE OF REPORT & PERIOD COVERED 9 Interim report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Robert Gary Goodman		8. CONTRACT OR GRANT NUMBER(s) 15 F44620-73-C-0074, ARPA Order-2446
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Computer Science Dept. Pittsburgh, PA 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101D AO 2446
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington, VA 22209		12. REPORT DATE 11 ~~May~~ Sep 1976
		13. NUMBER OF PAGES 169 12 171p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Air Force Office of Scientific Research (NM) Bolling AFB, DC 20332		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see attached

403 081

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)